

# Manual Técnico del Proyecto OakLand

**OakLand**, es un compilador basado en el lenguaje de programación Java, implementado utilizando JavaScript puro. La interfaz gráfica permite escribir, cargar y ejecutar archivos “.oak”, además de generar reportes de errores y de la tabla de símbolos.

## Características

- **JavaScript Puro:** El núcleo del proyecto está desarrollado completamente en JavaScript, sin el uso de frameworks o bibliotecas adicionales. Se utilizaron funciones nativas de JavaScript para gestionar la lógica del lenguaje.
- **PEG.js (Parser Expression Grammar):** Se utilizó PEG.js para la creación del parser. PEG.js es una herramienta que permite generar analizadores sintácticos a partir de gramáticas formales. En este caso, la gramática definida permite procesar las instrucciones del lenguaje OakLand, convertirlas en árboles de sintaxis abstractos (ASTs), y entregarlas al compilador para su ejecución.
- **Patrón Visitor:** Para el manejo del procesamiento de las instrucciones, se implementó el patrón de diseño Visitor, que permite ejecutar comportamientos sobre nodos del AST sin modificar la estructura de dichos nodos. Este patrón fue clave para gestionar:
  - **Variables:** Declaración, asignación, y manejo de tipos.
  - **Arreglos:** Inicialización, manipulación, y acceso a sus elementos.
  - **Sentencias:** Condicionales, bucles, y expresiones.
- **Generador de Instrucciones RISC-V:** Este componente traduce el código fuente a instrucciones de bajo nivel en la arquitectura RISC-V, optimizando el rendimiento y la ejecución en un entorno de hardware específico. El generador se encarga de transformar el AST en un conjunto de instrucciones que pueden ser ejecutadas directamente por un procesador RISC-V, aprovechando la eficiencia y la simplicidad de esta arquitectura. Además, se implementan mecanismos de optimización que permiten reducir la cantidad de instrucciones generadas y mejorar la ejecución del código, garantizando que las operaciones se realicen de manera efectiva y rápida.

## Estructura del proyecto

El proyecto está dividido en varios componentes clave:

- **Parser:** El archivo de gramática generado por PEG.js transforma el código fuente .oak en un AST. Este árbol es procesado por el compilador a través de visitas a cada nodo, aplicando la lógica definida en el patrón Visitor.
- **Compilador:** Implementado en JavaScript, este componente recibe el AST del parser y lo recorre utilizando el patrón Visitor. Aquí es donde se gestionan las declaraciones, asignaciones, y ejecución de las distintas instrucciones del lenguaje, traduciendo el código a instrucciones de RISC-V, esto se logra gracias a nuevos archivos, llamados generador, donde se encuentran instrucciones de RISC, las constantes de este lenguaje de bajo nivel además de builtins y utilidades auxiliares para el buen desarrollo del programa.
- **Interfaz Gráfica (HTML/CSS):** La interfaz fue desarrollada usando HTML y CSS, ofreciendo una experiencia visual simple pero funcional. El diseño fue optimizado para la entrada y salida de comandos, la carga de archivos, y la generación de reportes. CSS se usó para dar estilo a los botones, áreas de texto y consola, manteniendo una estética minimalista.

## Detalles de implementación

- **Manejo de Variables y Tipos:** Las variables pueden ser de diferentes tipos, como int, float, string, boolean, y char. Las variables se manejan dentro de un entorno dinámico que valida los tipos y sugiere errores cuando se intenta asignar un tipo incorrecto a una variable.
- **Arreglos:** Se implementaron arreglos con soporte para múltiples tipos. El sistema valida que los elementos dentro del arreglo sean consistentes con el tipo declarado, permitiendo manipular datos como listas o colecciones ordenadas.
- **Sentencias de Control:** Se implementaron las sentencias condicionales (if, else) y los ciclos (while, for), permitiendo la ejecución dinámica de bloques de código en función de ciertas condiciones o iteraciones.

## Despliegue

El proyecto está diseñado para ser desplegado utilizando **GitHub Pages**. Esto permite que la aplicación web sea accesible desde cualquier navegador sin necesidad de instalación local. El código fuente, incluyendo el parser, el compilador, y la interfaz, está completamente alojado en un repositorio de GitHub.

## Conclusión

El proyecto **OakLand** combina el poder de JavaScript puro con la flexibilidad de PEG.js y el patrón Visitor para ofrecer una experiencia completa de compilación de código a través de una interfaz web. El despliegue en GitHub Pages asegura que el proyecto sea fácilmente accesible y mantenible. Con la adición del generador de instrucciones RISC-V, OakLand no solo permite la compilación y ejecución de programas en un entorno web, sino que también proporciona una base sólida para la optimización y la ejecución de código en arquitecturas modernas, brindando a los desarrolladores un entorno robusto y eficiente para el desarrollo de software.

## Anexos:

- Captura del código de `generador.js` y `constantes.js` de RISC-V

```
25 export class Generador {
37   getLabel() {
38     return `L${this._labelCounter++}`
40   }
42   addLabel(label) {
43     label = label || this.getLabel()
44     this.instrucciones.push(new Instruction(`${label}:`))
45     return label
46   }
48   add(rd, rs1, rs2) {
49     this.instrucciones.push(new Instruction('add', rd, rs1, rs2))
50   }
52   sub(rd, rs1, rs2) {
53     this.instrucciones.push(new Instruction('sub', rd, rs1, rs2))
54   }
56   mul(rd, rs1, rs2) {
57     this.instrucciones.push(new Instruction('mul', rd, rs1, rs2))
58   }
60   div(rd, rs1, rs2) {
61     this.instrucciones.push(new Instruction('div', rd, rs1, rs2))
62   }
64   rem(rd, rs1, rs2) {
65     this.instrucciones.push(new Instruction('rem', rd, rs1, rs2))
66   }
68   addi(rd, rs1, inmediato) {
69     this.instrucciones.push(new Instruction('addi', rd, rs1, inmediato))
70   }
72   sw(rd, rs2, inmediato = 0) {
73     this.instrucciones.push(new Instruction('sw', rd, rs1, `${inmediato}${(rs2)}`))
74   }
76   sb(rd, rs2, inmediato = 0) {
77     this.instrucciones.push(new Instruction('sb', rd, rs1, `${inmediato}${(rs2)}`))
78   }
79   lw(rd, rs1, inmediato = 0) {
80     this.instrucciones.push(new Instruction('lw', rd, rs1, `${inmediato}${(rs1)}`))
81   }
83   lb(rd, rs1, inmediato = 0) {
84     this.instrucciones.push(new Instruction('lb', rd, rs1, `${inmediato}${(rs1)}`))
85   }
87 }

1 export const registers = [
2   28RD: 'zero',
3   30: 'ra',
4   31: 'sp',
5   32: 'gp',
6   33: 'tp',
7   34: 't0',
8   35: 't1',
9   36: 't2',
10  37: 't3',
11  38: 't4',
12  39: 't5',
13  40: 't6',
14  41: 'a0',
15  42: 'a1',
16  43: 'a2',
17  44: 'a3',
18  45: 'a4',
19  46: 'a5',
20  47: 'a6',
21  48: 'a7',
22  49: 's0',
23  50: 's1',
24  51: 's2',
25  52: 's3',
26  53: 's4',
27  54: 's5',
28  55: 's6',
29  56: 's7',
30  57: 's8',
31  58: 's9',
32  59: 's10',
33  60: 's11',
34  61: 'ra',
35  62: 'ra',
36  63: 'ra',
37  64: 'ra',
38  65: 'ra',
39  66: 'ra',
40  67: 'ra',
41  68: 'ra',
42  69: 'ra',
43  70: 'ra',
44  71: 'ra',
45  72: 'ra',
46  73: 'ra',
47  74: 'ra',
48  75: 'ra',
49  76: 'ra',
50  77: 'ra',
51  78: 'ra',
52  79: 'ra',
53  80: 'ra',
54  81: 'ra',
55  82: 'ra',
56  83: 'ra',
57  84: 'ra',
58  85: 'ra',
59  86: 'ra',
60  87: 'ra',
61  88: 'ra',
62  89: 'ra',
63  90: 'ra',
64  91: 'ra',
65  92: 'ra',
66  93: 'ra',
67  94: 'ra',
68  95: 'ra',
69  96: 'ra',
70  97: 'ra',
71  98: 'ra',
72  99: 'ra',
73  100: 'ra',
74  101: 'ra',
75  102: 'ra',
76  103: 'ra',
77  104: 'ra',
78  105: 'ra',
79  106: 'ra',
80  107: 'ra',
81  108: 'ra',
82  109: 'ra',
83  110: 'ra',
84  111: 'ra',
85  112: 'ra',
86  113: 'ra',
87  114: 'ra',
88  115: 'ra',
89  116: 'ra',
90  117: 'ra',
91  118: 'ra',
92  119: 'ra',
93  120: 'ra',
94  121: 'ra',
95  122: 'ra',
96  123: 'ra',
97  124: 'ra',
98  125: 'ra',
99  126: 'ra',
100 127: 'ra',
101 128: 'ra',
102 129: 'ra',
103 130: 'ra',
104 131: 'ra',
105 132: 'ra',
106 133: 'ra',
107 134: 'ra',
108 135: 'ra',
109 136: 'ra',
110 137: 'ra',
111 138: 'ra',
112 139: 'ra',
113 140: 'ra',
114 141: 'ra',
115 142: 'ra',
116 143: 'ra',
117 144: 'ra',
118 145: 'ra',
119 146: 'ra',
120 147: 'ra',
121 148: 'ra',
122 149: 'ra',
123 150: 'ra',
124 151: 'ra',
125 152: 'ra',
126 153: 'ra',
127 154: 'ra',
128 155: 'ra',
129 156: 'ra',
130 157: 'ra',
131 158: 'ra',
132 159: 'ra',
133 160: 'ra',
134 161: 'ra',
135 162: 'ra',
136 163: 'ra',
137 164: 'ra',
138 165: 'ra',
139 166: 'ra',
140 167: 'ra',
141 168: 'ra',
142 169: 'ra',
143 170: 'ra',
144 171: 'ra',
145 172: 'ra',
146 173: 'ra',
147 174: 'ra',
148 175: 'ra',
149 176: 'ra',
150 177: 'ra',
151 178: 'ra',
152 179: 'ra',
153 180: 'ra',
154 181: 'ra',
155 182: 'ra',
156 183: 'ra',
157 184: 'ra',
158 185: 'ra',
159 186: 'ra',
160 187: 'ra',
161 188: 'ra',
162 189: 'ra',
163 190: 'ra',
164 191: 'ra',
165 192: 'ra',
166 193: 'ra',
167 194: 'ra',
168 195: 'ra',
169 196: 'ra',
170 197: 'ra',
171 198: 'ra',
172 199: 'ra',
173 200: 'ra',
174 201: 'ra',
175 202: 'ra',
176 203: 'ra',
177 204: 'ra',
178 205: 'ra',
179 206: 'ra',
180 207: 'ra',
181 208: 'ra',
182 209: 'ra',
183 210: 'ra',
184 211: 'ra',
185 212: 'ra',
186 213: 'ra',
187 214: 'ra',
188 215: 'ra',
189 216: 'ra',
190 217: 'ra',
191 218: 'ra',
192 219: 'ra',
193 220: 'ra',
194 221: 'ra',
195 222: 'ra',
196 223: 'ra',
197 224: 'ra',
198 225: 'ra',
199 226: 'ra',
200 227: 'ra',
201 228: 'ra',
202 229: 'ra',
203 230: 'ra',
204 231: 'ra',
205 232: 'ra',
206 233: 'ra',
207 234: 'ra',
208 235: 'ra',
209 236: 'ra',
210 237: 'ra',
211 238: 'ra',
212 239: 'ra',
213 240: 'ra',
214 241: 'ra',
215 242: 'ra',
216 243: 'ra',
217 244: 'ra',
218 245: 'ra',
219 246: 'ra',
220 247: 'ra',
221 248: 'ra',
222 249: 'ra',
223 250: 'ra',
224 251: 'ra',
225 252: 'ra',
226 253: 'ra',
227 254: 'ra',
228 255: 'ra',
229 256: 'ra',
230 257: 'ra',
231 258: 'ra',
232 259: 'ra',
233 260: 'ra',
234 261: 'ra',
235 262: 'ra',
236 263: 'ra',
237 264: 'ra',
238 265: 'ra',
239 266: 'ra',
240 267: 'ra',
241 268: 'ra',
242 269: 'ra',
243 270: 'ra',
244 271: 'ra',
245 272: 'ra',
246 273: 'ra',
247 274: 'ra',
248 275: 'ra',
249 276: 'ra',
250 277: 'ra',
251 278: 'ra',
252 279: 'ra',
253 280: 'ra',
254 281: 'ra',
255 282: 'ra',
256 283: 'ra',
257 284: 'ra',
258 285: 'ra',
259 286: 'ra',
260 287: 'ra',
261 288: 'ra',
262 289: 'ra',
263 290: 'ra',
264 291: 'ra',
265 292: 'ra',
266 293: 'ra',
267 294: 'ra',
268 295: 'ra',
269 296: 'ra',
270 297: 'ra',
271 298: 'ra',
272 299: 'ra',
273 300: 'ra',
274 301: 'ra',
275 302: 'ra',
276 303: 'ra',
277 304: 'ra',
278 305: 'ra',
279 306: 'ra',
280 307: 'ra',
281 308: 'ra',
282 309: 'ra',
283 310: 'ra',
284 311: 'ra',
285 312: 'ra',
286 313: 'ra',
287 314: 'ra',
288 315: 'ra',
289 316: 'ra',
290 317: 'ra',
291 318: 'ra',
292 319: 'ra',
293 320: 'ra',
294 321: 'ra',
295 322: 'ra',
296 323: 'ra',
297 324: 'ra',
298 325: 'ra',
299 326: 'ra',
300 327: 'ra',
301 328: 'ra',
302 329: 'ra',
303 330: 'ra',
304 331: 'ra',
305 332: 'ra',
306 333: 'ra',
307 334: 'ra',
308 335: 'ra',
309 336: 'ra',
310 337: 'ra',
311 338: 'ra',
312 339: 'ra',
313 340: 'ra',
314 341: 'ra',
315 342: 'ra',
316 343: 'ra',
317 344: 'ra',
318 345: 'ra',
319 346: 'ra',
320 347: 'ra',
321 348: 'ra',
322 349: 'ra',
323 350: 'ra',
324 351: 'ra',
325 352: 'ra',
326 353: 'ra',
327 354: 'ra',
328 355: 'ra',
329 356: 'ra',
330 357: 'ra',
331 358: 'ra',
332 359: 'ra',
333 360: 'ra',
334 361: 'ra',
335 362: 'ra',
336 363: 'ra',
337 364: 'ra',
338 365: 'ra',
339 366: 'ra',
340 367: 'ra',
341 368: 'ra',
342 369: 'ra',
343 370: 'ra',
344 371: 'ra',
345 372: 'ra',
346 373: 'ra',
347 374: 'ra',
348 375: 'ra',
349 376: 'ra',
350 377: 'ra',
351 378: 'ra',
352 379: 'ra',
353 380: 'ra',
354 381: 'ra',
355 382: 'ra',
356 383: 'ra',
357 384: 'ra',
358 385: 'ra',
359 386: 'ra',
360 387: 'ra',
361 388: 'ra',
362 389: 'ra',
363 390: 'ra',
364 391: 'ra',
365 392: 'ra',
366 393: 'ra',
367 394: 'ra',
368 395: 'ra',
369 396: 'ra',
370 397: 'ra',
371 398: 'ra',
372 399: 'ra',
373 400: 'ra',
374 401: 'ra',
375 402: 'ra',
376 403: 'ra',
377 404: 'ra',
378 405: 'ra',
379 406: 'ra',
380 407: 'ra',
381 408: 'ra',
382 409: 'ra',
383 410: 'ra',
384 411: 'ra',
385 412: 'ra',
386 413: 'ra',
387 414: 'ra',
388 415: 'ra',
389 416: 'ra',
390 417: 'ra',
391 418: 'ra',
392 419: 'ra',
393 420: 'ra',
394 421: 'ra',
395 422: 'ra',
396 423: 'ra',
397 424: 'ra',
398 425: 'ra',
399 426: 'ra',
400 427: 'ra',
401 428: 'ra',
402 429: 'ra',
403 430: 'ra',
404 431: 'ra',
405 432: 'ra',
406 433: 'ra',
407 434: 'ra',
408 435: 'ra',
409 436: 'ra',
410 437: 'ra',
411 438: 'ra',
412 439: 'ra',
413 440: 'ra',
414 441: 'ra',
415 442: 'ra',
416 443: 'ra',
417 444: 'ra',
418 445: 'ra',
419 446: 'ra',
420 447: 'ra',
421 448: 'ra',
422 449: 'ra',
423 450: 'ra',
424 451: 'ra',
425 452: 'ra',
426 453: 'ra',
427 454: 'ra',
428 455: 'ra',
429 456: 'ra',
430 457: 'ra',
431 458: 'ra',
432 459: 'ra',
433 460: 'ra',
434 461: 'ra',
435 462: 'ra',
436 463: 'ra',
437 464: 'ra',
438 465: 'ra',
439 466: 'ra',
440 467: 'ra',
441 468: 'ra',
442 469: 'ra',
443 470: 'ra',
444 471: 'ra',
445 472: 'ra',
446 473: 'ra',
447 474: 'ra',
448 475: 'ra',
449 476: 'ra',
450 477: 'ra',
451 478: 'ra',
452 479: 'ra',
453 480: 'ra',
454 481: 'ra',
455 482: 'ra',
456 483: 'ra',
457 484: 'ra',
458 485: 'ra',
459 486: 'ra',
460 487: 'ra',
461 488: 'ra',
462 489: 'ra',
463 490: 'ra',
464 491: 'ra',
465 492: 'ra',
466 493: 'ra',
467 494: 'ra',
468 495: 'ra',
469 496: 'ra',
470 497: 'ra',
471 498: 'ra',
472 499: 'ra',
473 500: 'ra',
474 501: 'ra',
475 502: 'ra',
476 503: 'ra',
477 504: 'ra',
478 505: 'ra',
479 506: 'ra',
480 507: 'ra',
481 508: 'ra',
482 509: 'ra',
483 510: 'ra',
484 511: 'ra',
485 512: 'ra',
486 513: 'ra',
487 514: 'ra',
488 515: 'ra',
489 516: 'ra',
490 517: 'ra',
491 518: 'ra',
492 519: 'ra',
493 520: 'ra',
494 521: 'ra',
495 522: 'ra',
496 523: 'ra',
497 524: 'ra',
498 525: 'ra',
499 526: 'ra',
500 527: 'ra',
501 528: 'ra',
502 529: 'ra',
503 530: 'ra',
504 531: 'ra',
505 532: 'ra',
506 533: 'ra',
507 534: 'ra',
508 535: 'ra',
509 536: 'ra',
510 537: 'ra',
511 538: 'ra',
512 539: 'ra',
513 540: 'ra',
514 541: 'ra',
515 542: 'ra',
516 543: 'ra',
517 544: 'ra',
518 545: 'ra',
519 546: 'ra',
520 547: 'ra',
521 548: 'ra',
522 549: 'ra',
523 550: 'ra',
524 551: 'ra',
525 552: 'ra',
526 553: 'ra',
527 554: 'ra',
528 555: 'ra',
529 556: 'ra',
530 557: 'ra',
531 558: 'ra',
532 559: 'ra',
533 560: 'ra',
534 561: 'ra',
535 562: 'ra',
536 563: 'ra',
537 564: 'ra',
538 565: 'ra',
539 566: 'ra',
540 567: 'ra',
541 568: 'ra',
542 569: 'ra',
543 570: 'ra',
544 571: 'ra',
545 572: 'ra',
546 573: 'ra',
547 574: 'ra',
548 575: 'ra',
549 576: 'ra',
550 577: 'ra',
551 578: 'ra',
552 579: 'ra',
553 580: 'ra',
554 581: 'ra',
555 582: 'ra',
556 583: 'ra',
557 584: 'ra',
558 585: 'ra',
559 586: 'ra',
560 587: 'ra',
561 588: 'ra',
562 589: 'ra',
563 590: 'ra',
564 591: 'ra',
565 592: 'ra',
566 593: 'ra',
567 594: 'ra',
568 595: 'ra',
569 596: 'ra',
570 597: 'ra',
571 598: 'ra',
572 599: 'ra',
573 600: 'ra',
574 601: 'ra',
575 602: 'ra',
576 603: 'ra',
577 604: 'ra',
578 605: 'ra',
579 606: 'ra',
580 607: 'ra',
581 608: 'ra',
582 609: 'ra',
583 610: 'ra',
584 611: 'ra',
585 612: 'ra',
586 613: 'ra',
587 614: 'ra',
588 615: 'ra',
589 616: 'ra',
590 617: 'ra',
591 618: 'ra',
592 619: 'ra',
593 620: 'ra',
594 621: 'ra',
595 622: 'ra',
596 623: 'ra',
597 624: 'ra',
598 625: 'ra',
599 626: 'ra',
600 627: 'ra',
601 628: 'ra',
602 629: 'ra',
603 630: 'ra',
604 631: 'ra',
605 632: 'ra',
606 633: 'ra',
607 634: 'ra',
608 635: 'ra',
609 636: 'ra',
610 637: 'ra',
611 638: 'ra',
612 639: 'ra',
613 640: 'ra',
614 641: 'ra',
615 642: 'ra',
616 643: 'ra',
617 644: 'ra',
618 645: 'ra',
619 646: 'ra',
620 647: 'ra',
621 648: 'ra',
622 649: 'ra',
623 650: 'ra',
624 651: 'ra',
625 652: 'ra',
626 653: 'ra',
627 654: 'ra',
628 655: 'ra',
629 656: 'ra',
630 657: 'ra',
631 658: 'ra',
632 659: 'ra',
633 660: 'ra',
634 661: 'ra',
635 662: 'ra',
636 663: 'ra',
637 664: 'ra',
638 665: 'ra',
639 666: 'ra',
640 667: 'ra',
641 668: 'ra',
642 669: 'ra',
643 670: 'ra',
644 671: 'ra',
645 672: 'ra',
646 673: 'ra',
647 674: 'ra',
648 675: 'ra',
649 676: 'ra',
650 677: 'ra',
651 678: 'ra',
652 679: 'ra',
653 680: 'ra',
654 681: 'ra',
655 682: 'ra',
656 683: 'ra',
657 684: 'ra',
658 685: 'ra',
659 686: 'ra',
660 687: 'ra',
661 688: 'ra',
662 689: 'ra',
663 690: 'ra',
664 691: 'ra',
665 692: 'ra',
666 693: 'ra',
667 694: 'ra',
668 695: 'ra',
669 696: 'ra',
670 697: 'ra',
671 698: 'ra',
672 699: 'ra',
673 700: 'ra',
674 701: 'ra',
675 702: 'ra',
676 703: 'ra',
677 704: 'ra',
678 705: 'ra',
679 706: 'ra',
680 707: 'ra',
681 708: 'ra',
682 709: 'ra',
683 710: 'ra',
684 711: 'ra',
685 712: 'ra',
686 713: 'ra',
687 714: 'ra',
688 715: 'ra',
689 716: 'ra',
690 717: 'ra',
691 718: 'ra',
692 719: 'ra',
693 720: 'ra',
694 721: 'ra',
695 722: 'ra',
696 723: 'ra',
697 724: 'ra',
698 725: 'ra',
699 726: 'ra',
700 727: 'ra',
701 728: 'ra',
702 729: 'ra',
703 730: 'ra',
704 731: 'ra',
705 732: 'ra',
706 733: 'ra',
707 734: 'ra',
708 735: 'ra',
709 736: 'ra',
710 737: 'ra',
711 738: 'ra',
712 739: 'ra',
713 740: 'ra',
714 741: 'ra',
715 742: 'ra',
716 743: 'ra',
717 744: 'ra',
718 745: 'ra',
719 746: 'ra',
720 747: 'ra',
721 748: 'ra',
722 749: 'ra',
723 750: 'ra',
724 751: 'ra',
725 752: 'ra',
726 753: 'ra',
727 754: 'ra',
728 755: 'ra',
729 756: 'ra',
730 757: 'ra',
731 758: 'ra',
732 759: 'ra',
733 760: 'ra',
734 761: 'ra',
735 762: 'ra',
736 763: 'ra',
737 764: 'ra',
738 765: 'ra',
739 766: 'ra',
740 767: 'ra',
741 768: 'ra',
742 769: 'ra',
743 770: 'ra',
744 771: 'ra',
745 772: 'ra',
746 773: 'ra',
747 774: 'ra',
748 775: 'ra',
749 776: 'ra',
750 777: 'ra',
751 778: 'ra',
752 779: 'ra',
753 780: 'ra',
754 781: 'ra',
755 782: 'ra',
756 783: 'ra',
757 784: 'ra',
758 785: 'ra',
759 786: 'ra',
760 787: 'ra',
761 788: 'ra',
762 789: 'ra',
763 790: 'ra',
764 791: 'ra',
765 792: 'ra',
766 793: 'ra',
767 794: 'ra',
768 795: 'ra',
769 796: 'ra',
770 797: 'ra',
771 798: 'ra',
772 799: 'ra',
773 800: 'ra',
774 801: 'ra',
775 802: 'ra',
776 803: 'ra',
777 804: 'ra',
778 805: 'ra',
779 806: 'ra',
780 807: 'ra',
781 808: 'ra',
782 809: 'ra',
783 810: 'ra',
784 811: 'ra',
785 812: 'ra',
786 813: 'ra',
787 814: 'ra',
788 815: 'ra',
789 816: 'ra',
790 817: 'ra',
791 818: 'ra',
792 819: 'ra',
793 820: 'ra',
794 821: 'ra',
795 822: 'ra',
796 823: 'ra',
797 824: 'ra',
798 825: 'ra',
799 826: 'ra',
800 827: 'ra',
801 828: 'ra',
802 829: 'ra',
803 830: 'ra',
804 831: 'ra',
805 832: 'ra',
806 833: 'ra',
807 834: 'ra',
808 835: 'ra',
809 836: 'ra',
810 837: 'ra',
811 838: 'ra',
812 839: 'ra',
813 840: 'ra',
814 841: 'ra',
815 842: 'ra',
816 843: 'ra',
817 844: 'ra',
818 845: 'ra',
819 846: 'ra',
820 847: 'ra',
821 848: 'ra',
822 849: 'ra',
823 850: 'ra',
824 851: 'ra',
825 852: 'ra',
826 853: 'ra',
827 854: 'ra',
828 855: 'ra',
829 856: 'ra',
830 857: 'ra',
831 858: 'ra',
832 859: 'ra',
833 860: 'ra',
834 861: 'ra',
835 862: 'ra',
836 863: 'ra',
837 864: 'ra',
838 865: 'ra',
839 866: 'ra',
840 867: 'ra',
841 868: 'ra',
842 869: 'ra',
843 870: 'ra',
844 871: 'ra',
845 872: 'ra',
846 873: 'ra',
847 874: 'ra',
848 875: 'ra',
849 876: 'ra',
850 877: 'ra',
851 878: 'ra',
852 879: 'ra',
853 880: 'ra',
854 881: 'ra',
855 882: 'ra',
856 883: 'ra',
857 884: 'ra',
858 885: 'ra',
859 886: 'ra',
860 887: 'ra',
861 888: 'ra',
862 889: 'ra',
863 890: 'ra',
864 891: 'ra',
865 892: 'ra',
866 893: 'ra',
867 894: 'ra',
868 895: 'ra',
869 896: 'ra',
870 897: 'ra',
871 898: 'ra',
872 899: 'ra',
873 900: 'ra',
874 901: 'ra',
875 902: 'ra',
876 903: 'ra',
877 904: 'ra',
878 905: 'ra',
879 906: 'ra',
880 907: 'ra',
881 908: 'ra',
882 909: 'ra',
883 910: 'ra',
884 911: 'ra',
885 912: 'ra',
886 913: 'ra',
887 914: 'ra',
888 915: 'ra',
889 916: 'ra',
890 917: 'ra',
891 918: 'ra',
892 919: 'ra',
893 920: 'ra',
894 921: 'ra',
895 922: 'ra',
896 923: 'ra',
897 924: 'ra',
898 925: 'ra',
899 926: 'ra',
900 927: 'ra',
901 928: 'ra',
902 929: 'ra',
903 930: 'ra',
904 931: 'ra',
905 932: 'ra',
906 933: 'ra',
907 934: 'ra',
908 935: 'ra',
909 936: 'ra',
910 937: 'ra',
911 938: 'ra',
912 939: 'ra',
913 940: 'ra',
914 941: 'ra',
915 942: 'ra',
916 943: 'ra',
917 944: 'ra',
918 945: 'ra',
919 946: 'ra',
920 947: 'ra',
921 948: 'ra',
922 949: 'ra',
923 950: 'ra',
924 951: 'ra',
925 952: 'ra',
926 953: 'ra',
927 954: 'ra',
928 955: 'ra',
929 956: 'ra',
930 957: '
```

- Captura del código de Compiler Visitor

```

import class CompilerVisitor extends BaseVisitor {
    constructor() {
        super()
        this.code = new Generator()
        this.continueLabel = null
        this.breakLabel = null

        this.functionSetada = {}
        this.isInFunction = false
        this.frameIndices = 0
        this.returnLabel = null
    }

    /**
     * @type (BaseVisitor["visitExpressionInt"])
     */
    visitExpressionInt(node) {
        node.exp.accept(this)
        this.code.popObject(r, 10)
    }

    /**
     * @type (BaseVisitor["visitPrimitive"])
     */
    visitPrimitive(node) {
        this.code.ll(r, 10, node.valor)
        this.code.push()
        this.code.comment("Primitive: ${node.valor}")
        this.code.pushConstant({ type: node.tipo, valor: node.valor })
        this.code.comment("fin Primitive: ${node.valor}")
    }

    /**
     * @type (BaseVisitor["visitOperacionBinaria"])
     */
    visitOperacionBinaria(node) {
        this.code.comment("Operacion: ${node.op}")

        if (node.op === "==" || "!=") {
            node.lq.accept(this) // lq
            this.code.popObject(r, 10) // lq

            const labelFalse = this.code.getLabel()
            const labelEnd = this.code.getLabel()

            this.code.beq(r, 10, r, ZERO, labelFalse) // if(!lq) goto labelFalse
            node.der.accept(this) // der
        }
    }
}

```

- Interfaz gráfica traduciendo instrucciones de lenguaje de alto nivel a RISC-V

