



**“AÑO DE LA RECUPERACIÓN Y CONSOLIDACIÓN DE LA
ECONOMÍA PERUANA”**

TÍTULO:

SISTEMA DE GESTIÓN DE PROCESOS

CURSO:

ESTRUCTURA DE DATOS

DOCENTE:

Osorio Contreras, Rosario Delia

INTEGRANTES:

- Capani Paitan Brayan Ronaldo
- Izarra Flores Jewilson Jenkins
- Penadillo Gonzales Gilmer Antonny
- Antezana Alonzo Alexandra Dayana

HUANCAYO

2025

CAPÍTULO 1: Análisis del Problema:

1. Descripción del Problema

El sistema operativo de cualquier computadora requiere gestionar múltiples procesos de manera eficiente. Para simular este entorno, se propone el desarrollo de un “Sistema de Gestión de Procesos” que permita al usuario agregar, modificar, eliminar, ejecutar y administrar procesos de forma dinámica, utilizando exclusivamente estructuras de datos lineales dinámicas.

Este proyecto busca resolver el problema de la administración de procesos mediante un entorno educativo simulado, donde se pueden registrar procesos, asignarles prioridades, ejecutarlos según esa prioridad y simular la asignación de memoria, todo esto con estructuras diseñadas e implementadas manualmente.

El desarrollo de este sistema permite aplicar los conocimientos de la Unidad 3 del curso "Estructura de Datos", reforzando la comprensión del uso práctico de listas enlazadas, colas de prioridad y pilas. A su vez, fomenta el trabajo colaborativo, la documentación técnica y el control de versiones mediante el uso de herramientas digitales como GitHub.

Requerimientos funcionales

- .- Insertar nuevos procesos (con ID, nombre y prioridad).
- .- Mostrar la lista de procesos actuales.
- .- Buscar procesos por ID.
- .- Modificar la prioridad de un proceso.
- .- Eliminar un proceso específico.
- .- Encolar procesos para ejecución según su prioridad.
- .- Desencolar (ejecutar) el proceso más prioritario.
- .- Mostrar el contenido de la cola de prioridad.
- .- Asignar memoria a un proceso (operación push en una pila).
- .- Liberar memoria (operación pop).
- .- Visualizar el estado actual de la memoria.
- .- Guardar los procesos en un archivo de texto ([procesos.txt](#)).
- .- Cargar automáticamente los procesos al iniciar el sistema.

Requisitos no funcionales

- Interfaz de usuario basada en consola, sencilla y funcional.
- Código estructurado en lenguaje C ++, ejecutable en Dev-C ++.
- Implementación de las estructuras sin el uso de bibliotecas predefinidas como `std::list`, `std::queue` o `std::stack`.
- Persistencia mediante archivos de texto sin uso de bases de datos.
- Control de versiones a través de un repositorio público en GitHub.
- Estructura de código modular, con comentarios descriptivos para facilitar su comprensión y mantenimiento.

3. Estructuras de datos propuestas

Para cumplir con los requisitos del sistema, se han definido las siguientes estructuras dinámicas:

- Lista enlazada simple: utilizada para mantener el registro de todos los procesos en memoria principal.
- Cola de prioridad: diseñada para gestionar la ejecución de procesos en orden de prioridad (prioridad más alta se atiende primero).
- Pila: empleada para simular la gestión de memoria temporal, asignando y liberando memoria a los procesos según el orden LIFO.

4. Justificación de la elección

Las estructuras seleccionadas se ajustan perfectamente a la lógica de un sistema operativo simplificado:

- La lista enlazada permite el manejo dinámico de procesos sin un tamaño predefinido, facilitando inserciones y eliminaciones en cualquier posición.
- La cola de prioridad es idónea para simular la planificación de tareas en una CPU, permitiendo que los procesos con mayor prioridad se ejecuten primero.
- La pila es ideal para representar el uso de memoria temporal, tal como ocurre en la ejecución real de funciones y subprocesos.

Estas estructuras fueron implementadas desde cero, manipulando punteros manualmente, lo que favorece el aprendizaje profundo de estructuras dinámicas en C + +, tal como lo exige la competencia del curso.

Capítulo 2: Diseño de la Solución:

2.1. Descripción de estructuras de datos y operaciones

Lista Enlazada (Gestor de Procesos)

- Operaciones: insertar, eliminar, buscar, modificar prioridad, mostrar procesos.
- Estructura: cada nodo contiene ID, nombre, prioridad y puntero al siguiente.

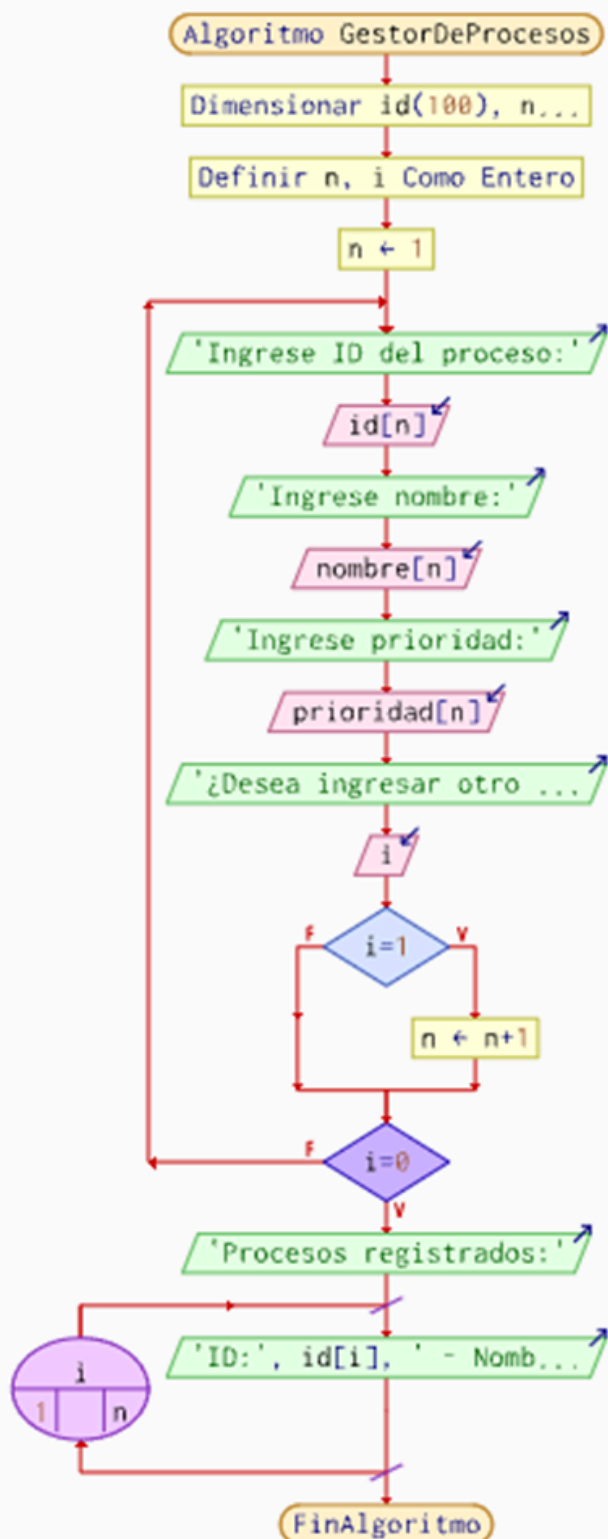
Cola de Prioridad (Planificador de CPU)

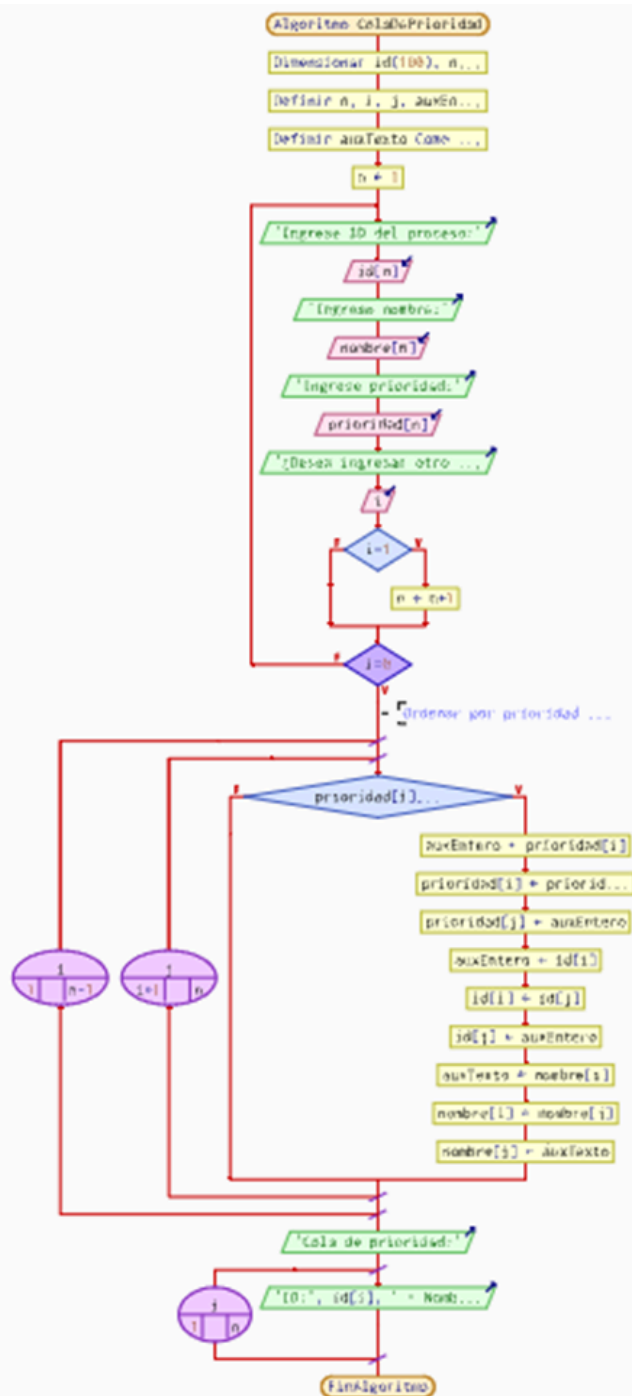
- Operaciones: encolar, desencolar, mostrar cola.
- Estructura: ordenada manualmente al insertar, según mayor prioridad.

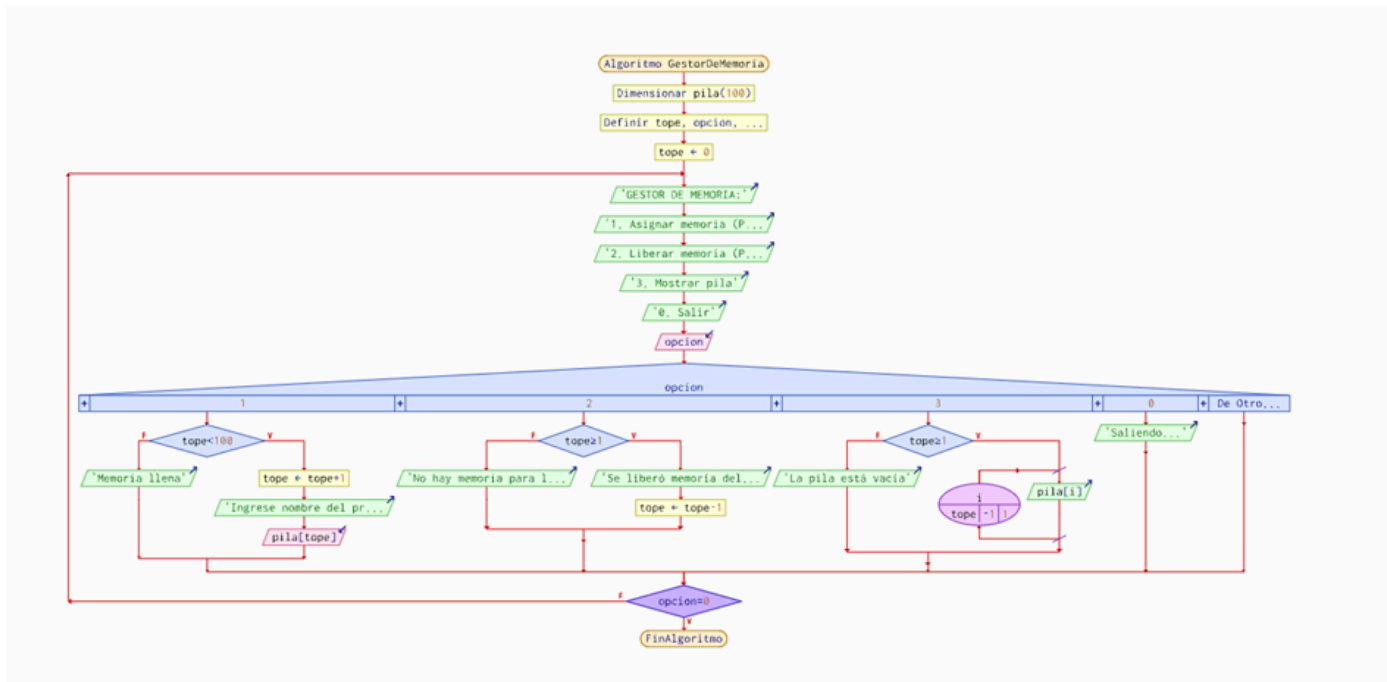
Pila (Gestor de Memoria)

- Operaciones: push (asignar memoria), pop (liberar memoria), mostrar memoria.
- Estructura: nodos apilados con puntero al siguiente (orden LIFO).

2.2. Diagramas de flujo:







2.3. Justificación del diseño

- Las estructuras dinámicas permiten trabajar con memoria flexible, sin tamaño fijo.
- Se simula una CPU y gestión de memoria real, sin usar bibliotecas avanzadas.
- La lista enlazada facilita inserciones/eliminaciones sin copiar datos.
- La cola de prioridad refuerza la lógica de ejecución en orden de importancia.
- La pila permite simular liberación de recursos de forma controlada (LIFO).
- El diseño modular del código mejora su comprensión, mantenimiento y evaluación por cada estructura.

CAPÍTULO 3: SOLUCIÓN FINAL:

3.1 Capturas de Pantalla:

Gestión de lista de procesos:

Insertar proceso:

```
Procesos cargados desde procesos.txt

===== MENÚ PRINCIPAL =====
1. Gestión de lista de procesos
2. Gestión de cola de prioridad (CPU)
3. Gestión de memoria (pila)
0. Guardar y salir
Elegir opción (0 - 3): 1

-- Gestión de Lista de Procesos --
1. Insertar proceso
2. Buscar proceso
3. Eliminar proceso
4. Modificar prioridad
5. Mostrar procesos
0. Volver al menú principal
Elegir opción (0 - 5): 1

Ingrese ID del proceso (número entero positivo): 45
Ingrese nombre del proceso: Facebook
Ingrese prioridad del proceso (mayor número = mayor prioridad, mínimo 0): 7
Proceso registrado exitosamente.
```

Mostrar procesos:

```
-- Gestión de Lista de Procesos --
1. Insertar proceso
2. Buscar proceso
3. Eliminar proceso
4. Modificar prioridad
5. Mostrar procesos
0. Volver al menú principal
Elegir opción (0 - 5): 5

Lista de procesos:
ID: 45 | Nombre: Facebook | Prioridad: 7
ID: 34 | Nombre: Trello | Prioridad: 8
ID: 78 | Nombre: Youtube | Prioridad: 57
ID: 23 | Nombre: Word | Prioridad: 14
```

Buscar proceso:

```
-- Gestión de Lista de Procesos --
1. Insertar proceso
2. Buscar proceso
3. Eliminar proceso
4. Modificar prioridad
5. Mostrar procesos
0. Volver al menú principal
Elegir opción (0 - 5): 2

Ingrese el ID del proceso a buscar: 23
Proceso encontrado:
ID: 23
Nombre: Word
Prioridad: 14
```


Eliminar proceso:

```
-- Gestión de Lista de Procesos --
1. Insertar proceso
2. Buscar proceso
3. Eliminar proceso
4. Modificar prioridad
5. Mostrar procesos
0. Volver al menú principal
Elegir opción (0 - 5): 3

Ingrese el ID del proceso a eliminar: 45
Proceso con ID 45 eliminado correctamente.

-- Gestión de Lista de Procesos --
1. Insertar proceso
2. Buscar proceso
3. Eliminar proceso
4. Modificar prioridad
5. Mostrar procesos
0. Volver al menú principal
Elegir opción (0 - 5): 5

Lista de procesos:
ID: 34 | Nombre: Trello | Prioridad: 8
ID: 78 | Nombre: Youtube | Prioridad: 57
ID: 23 | Nombre: Word | Prioridad: 14
```

Modificar prioridad:

```
1. Insertar proceso
2. Buscar proceso
3. Eliminar proceso
4. Modificar prioridad
5. Mostrar procesos
0. Volver al menú principal
Elegir opción (0 - 5): 4

Ingrese el ID del proceso a modificar: 34
Ingrese la nueva prioridad (mayor número = mayor prioridad, mínimo 0): 12

Proceso encontrado:
ID: 34 | Nombre: Trello | Prioridad actual: 8
Prioridad actualizada a: 12

-- Gestión de Lista de Procesos --
1. Insertar proceso
2. Buscar proceso
3. Eliminar proceso
4. Modificar prioridad
5. Mostrar procesos
0. Volver al menú principal
Elegir opción (0 - 5): 5

Lista de procesos:
ID: 34 | Nombre: Trello | Prioridad: 12
ID: 78 | Nombre: Youtube | Prioridad: 57
ID: 23 | Nombre: Word | Prioridad: 14
```

Gestión de cola de prioridad:

Encolar proceso:

```
===== MENÚ PRINCIPAL =====
1. Gestión de lista de procesos
2. Gestión de cola de prioridad (CPU)
3. Gestión de memoria (pila)
0. Guardar y salir
Elegir opción (0 - 3): 2

-- Gestión de Cola de Prioridad (CPU) --
1. Encolar proceso
2. Ejecutar proceso (desencolar)
3. Mostrar cola
0. Volver al menú principal
Elegir opción (0 - 3): 1

Ingrese ID del proceso a encolar: 34
Proceso 'Trello' encolado según prioridad.
```

Mostrar cola:

```
-- Gestión de Cola de Prioridad (CPU) --
1. Encolar proceso
2. Ejecutar proceso (desencolar)
3. Mostrar cola
0. Volver al menú principal
Elegir opción (0 - 3): 3

Cola de procesos por prioridad:
ID: 78 | Nombre: Youtube | Prioridad: 57
ID: 23 | Nombre: Word | Prioridad: 14
ID: 34 | Nombre: Trello | Prioridad: 12
```

Desencolar proceso:

```
-- Gestión de Cola de Prioridad (CPU) --
1. Encolar proceso
2. Ejecutar proceso (desencolar)
3. Mostrar cola
0. Volver al menú principal
Elegir opción (0 - 3): 2

Proceso ejecutado: Youtube (ID: 78)

-- Gestión de Cola de Prioridad (CPU) --
1. Encolar proceso
2. Ejecutar proceso (desencolar)
3. Mostrar cola
0. Volver al menú principal
Elegir opción (0 - 3): 3

Cola de procesos por prioridad:
ID: 23 | Nombre: Word | Prioridad: 14
ID: 34 | Nombre: Trello | Prioridad: 12
```

Gestión de memoria:

Asignar memoria:

```
===== MENÚ PRINCIPAL =====
1. Gestión de lista de procesos
2. Gestión de cola de prioridad (CPU)
3. Gestión de memoria (pila)
0. Guardar y salir
Elegir opción (0 - 3): 3

-- Gestión de Memoria (Pila) --
1. Asignar memoria (Push)
2. Liberar memoria (Pop)
3. Mostrar estado de la memoria
0. Volver al menú principal
Elegir opción (0 - 3): 1

Ingrese ID del proceso que usará memoria: 23
Ingrese el tamaño de memoria a asignar (en KB): 90
Memoria asignada al proceso 'Word' (90 KB).
```

Mostrar estado de memoria:

```
-- Gestión de Memoria (Pila) --
1. Asignar memoria (Push)
2. Liberar memoria (Pop)
3. Mostrar estado de la memoria
0. Volver al menú principal
Elegir opción (0 - 3): 3

Estado de la memoria (pila):
Proceso: Trello | Memoria asignada: 65 KB
Proceso: Word | Memoria asignada: 90 KB
```

Liberar memoria:

```
-- Gestión de Memoria (Pila) --
1. Asignar memoria (Push)
2. Liberar memoria (Pop)
3. Mostrar estado de la memoria
0. Volver al menú principal
Elegir opción (0 - 3): 2

Memoria liberada del proceso: Trello (65 KB)

-- Gestión de Memoria (Pila) --
1. Asignar memoria (Push)
2. Liberar memoria (Pop)
3. Mostrar estado de la memoria
0. Volver al menú principal
Elegir opción (0 - 3): 3

Estado de la memoria (pila):
Proceso: Word | Memoria asignada: 90 KB
```

3.2 Manual de Usuario del Sistema:

Requisitos:

- Sistema operativo con Dev-C ++ instalado.
- Archivo **Proyecto.cpp** compilado correctamente.

Pasos de uso:

1. **Ejecutar el programa desde Dev-C ++.**
2. **Se presentará un menú con 4 opciones principales:**
 - Gestión de lista de procesos
 - Cola de prioridad (CPU)
 - Gestión de memoria (pila)
 - Guardar y salir
3. **Ingresar la opción deseada (número entre 0 y 3).**
4. **En cada submenú, seguir las instrucciones para:**
 - Agregar, buscar, modificar o eliminar procesos.
 - Encolar y ejecutar procesos por prioridad.
 - Asignar o liberar memoria.
5. **Al finalizar, elegir la opción “Guardar y salir” para guardar los datos en el archivo **procesos.txt**.**

Notas:

- Si se intenta agregar un proceso con un ID repetido, el sistema lo rechazará.
- La prioridad más alta (número mayor) será ejecutada primero en la cola.
- El uso de memoria sigue el orden LIFO (último en entrar, primero en salir)

Capítulo 4: Evidencias de Trabajo en Equipo:

4.1. Repositorio con Control de Versiones (Capturas de Pantalla)

Update Proyecto.cpp
AlexandraAntezana • 10 minutos ago

entrega(actualizacion del codigo)
AlexandraAntezana • 18 minutos ago

Entrega de (mejora de la función lista d...
wilsonzara • 2 hours ago

entrega (función búsqueda por nombre)
Brayan • 3 hours ago

entrega Gilmer
Antonyy • 6 hours ago

Update Proyecto.cpp
Brayan09r • 7 days ago

Update Proyecto.cpp
wilsonzara • 7 days ago

Update Proyecto.cpp
wilsonzara • 7 days ago

Update Proyecto.cpp
Gilmer • 7 days ago

Update Proyecto.cpp
Gilmer • 7 days ago

Update Proyecto.cpp
AlexandraAntezana • 7 days ago

Update Proyecto.cpp

2 changed files

Proyecto.cpp

Proyecto.exe

Proyecto.cpp

@@ -4,8 +4,6 @@

4 4

5 5 using namespace std;

6 6

7 7 -

8 8 -

9 7 // ===== ESTRUCTURAS =====

10 8

11 9 // Nodo para lista enlazada (Gestor de procesos)

12 10

13 11 @@ -19,6 +17,7 @@ struct Nodo {

14 12

15 13 // Nodo para pila (Gestor de memoria)

16 14

17 15 struct NodoPila {

18 16

19 17 string proceso;

20 18

21 19 int tamanomemoria; //EN KB

22 20

23 21 NodoPila *siguiente;

24 22

25 23

26 24

27 25 @@ -27,8 +26,9 @@ Nodo *inicio = NULL; // Lista enlazada de procesos

28 26

29 27 Nodo *colaPrioridad = NULL; // Cola de prioridad (ordenada por prioridad)

30 28

31 29 NodoPila *tope = NULL; // Pila para memoria

32 30

33 31 - // ===== FUNCIONES DE VALIDACIÓN =====

34 32

35 33 + // ===== FUNCIONES DE VALIDACIÓN =====

36 34

37 35 // Lee un entero no negativo

38 36

39 37

40 38

41 39

42 40

43 41

44 42

45 43

46 44

47 45

48 46

49 47

50 48

51 49

52 50

53 51

54 52

55 53

56 54

57 55

58 56

59 57

60 58

61 59

62 60

63 61

64 62

65 63

66 64

67 65

68 66

69 67

70 68

71 69

72 70

73 71

74 72

75 73

76 74

77 75

78 76

79 77

80 78

81 79

82 80

83 81

84 82

85 83

86 84

87 85

88 86

89 87

90 88

91 89

92 90

93 91

94 92

95 93

96 94

97 95

98 96

99 97

100 98

101 99

102 100

103 101

104 102

105 103

106 104

107 105

108 106

109 107

110 108

111 109

112 110

113 111

114 112

115 113

116 114

117 115

118 116

119 117

120 118

121 119

122 120

123 121

124 122

125 123

126 124

127 125

128 126

129 127

130 128

131 129

132 130

133 131

134 132

135 133

136 134

137 135

138 136

139 137

140 138

141 139

142 140

143 141

144 142

145 143

146 144

147 145

148 146

149 147

150 148

151 149

152 150

153 151

154 152

155 153

156 154

157 155

158 156

159 157

160 158

161 159

162 160

163 161

164 162

165 163

166 164

167 165

168 166

169 167

170 168

171 169

172 170

173 171

174 172

175 173

176 174

177 175

178 176

179 177

180 178

181 179

182 180

183 181

184 182

185 183

186 184

187 185

188 186

189 187

190 188

191 189

192 190

193 191

194 192

195 193

196 194

197 195

198 196

199 197

200 198

201 199

202 200

203 201

204 202

205 203

206 204

207 205

208 206

209 207

210 208

211 209

212 210

213 211

214 212

215 213

216 214

217 215

218 216

219 217

220 218

221 219

222 220

223 221

224 222

225 223

226 224

227 225

228 226

229 227

230 228

231 229

232 230

233 231

234 232

235 233

236 234

237 235

238 236

239 237

240 238

241 239

242 240

243 241

244 242

245 243

246 244

247 245

248 246

249 247

250 248

251 249

252 250

253 251

254 252

255 253

256 254

257 255

258 256

259 257

260 258

261 259

262 260

263 261

264 262

265 263

266 264

267 265

268 266

269 267

270 268

271 269

272 270

273 271

274 272

275 273

276 274

277 275

278 276

279 277

280 278

281 279

282 280

283 281

284 282

285 283

286 284

287 285

288 286

289 287

290 288

291 289

292 290

293 291

294 292

295 293

296 294

297 295

298 296

299 297

300 298

301 299

302 300

303 301

304 302

305 303

306 304

307 305

308 306

309 307

310 308

311 309

312 310

313 311

314 312

315 313

316 314

317 315

318 316

319 317

320 318

321 319

322 320

323 321

324 322

325 323

326 324

327 325

328 326

329 327

330 328

331 329

332 330

333 331

334 332

335 333

336 334

337 335

338 336

339 337

340 338

341 339

342 340

343 341

344 342

345 343

346 344

347 345

348 346

349 347

350 348

351 349

352 350

353 351

354 352

355 353

356 354

357 355

358 356

359 357

360 358

361 359

362 360

363 361

364 362

365 363

366 364

367 365

368 366

369 367

370 368

371 369

372 370

373 371

374 372

375 373

376 374

377 375

378 376

379 377

380 378

381 379

382 380

383 381

384 382

385 383

386 384

387 385

388 386

389 387

390 388

391 389

392 390

393 391

394 392

395 393

396 394

397 395

398 396

399 397

400 398

401 399

402 400

403 401

404 402

405 403

406 404

407 405

408 406

409 407

410 408

411 409

412 410

413 411

414 412

415 413

416 414

417 415

418 416

419 417

420 418

421 419

422 420

423 421

424 422

425 423

426 424

427 425

428 426

429 427

430 428

431 429

432 430

433 431

434 432

435 433

436 434

437 435

438 436

439 437

440 438

441 439

442 440

443 441

444 442

445 443

446 444

447 445

448 446

449 447

450 448

451 449

452 450

453 451

454 452

455 453

456 454

457 455

458 456

459 457

460 458

461 459

462 460

463 461

464 462

465 463

466 464

467 465

468 466

469 467

470 468

471 469

472 470

473 471

474 472

475 473

476 474

477 475

478 476

479 477

480 478

481 479

482 480

483 481

484 482

485 483

486 484

487 485

488 486

489 487

490 488

491 489

492 490

493 491

494 492

495 493

496 494

497 495

498 496

499 497

500 498

501 499

502 500

503 501

504 502

505 503

506 504

507 505

508 506

509 507

510 508

511 509

512 510

513 511

514 512

515 513

516 514

517 515

518 516

519 517

520 518

521 519

522 520

523 521

524 522

525 523

526 524

527 525

528 526

529 527

530 528

531 529

532 530

533 531

534 532

535 533

536 534

537 535

538 536

539 537

540 538

541 539

542 540

543 541

544 542

545 543

546 544

547 545

548 546

549 547

550 548

551 549

552 550

553 551

554 552

555 553

556 554

557 555

558 556

559 557

560 558

561 559

562 560

563 561

564 562

565 563

566 564

567 565

568 566

569 567

570 568

571 569

572 570

573 571

574 572

575 573

576 574

577 575

578 576

579 577

580 578

581 579

582 580

583 581

584 582

585 583

586 584

587 585

588 586

589 587

590 588

591 589

592 590

593 591

594 592

595 593

596 594

597 595

598 596

599 597

600 598

601 599

602 600

603 601

604 602

605 603

606 604

607 605

608 606

609 607

610 608

611 609

612 610

613 611

614 612

615 613

616 614

617 615

618 616

619 617

620 618

621 619

622 620

623 621

624 622

625 623

626 624

627 625

628 626

629 627

630 628

631 629

632 630

633 631

634 632

635 633

636 634

637 635

638 636

639 637

640 638

641 639

642 640

643 641

644 642

645 643

646 644

647 645

648 646

649 647

650 648

651 649

652 650

653 651

654 652

655 653

656 654

657 655

658 656

659 657

660 658

661 659

662 660

663 661

664 662

665 663

666 664

667 665

668 666

669 667

670 668

671 669

672 670

673 671

674 672

675 673

676 674

677 675

678 676

679 677

680 678

681 679

682 680

683 681

684 682

685 683

686 684

687 685

688 686

689 687

690 688

691 689

692 690

693 691

694 692

695 693

696 694

697 695

698 696

699 697

700 698

701 699

702 700

703 701

704 702

705 703

706 704

707 705

708 706

709 707

710 708

711 709

712 710

713 711

714 712

715 713

716 714

717 715

718 716

719 717

720 718

721 719

722 720

723 721

724 722

725 723

726 724

727 725

728 726

729 727

730 728

731 729

732 730

733 731

734 732

735 733

736 734

737 735

738 736

739 737

740 738

741 739

742 740

743 741

744 742

745 743

746 744

747 745

748 746

749 747

750 748

751 749

752 750

753 751

754 752

755 753

756 754

757 755

758 756

759 757

760 758

761 759

762 760

763 761

764 762

765 763

766 764

767 765

768 766

769 767

770 768

771 769

772 770

773 771

774 772

775 773

776 774

777 775

778 776

779 777

780 778

781 779

782 780

783 781

784 782

785 783

786 784

787 785

788 786

789 787

790 788

791 789

792 790

793 791

794 792

795 793

796 794

797 795

798 796

799 797

800 798

801 799

802 800

803 801

804 802

805 803

806 804

807 805

808 806

809 807

810 808

811 809

812 810

813 811

814 812

815 813

816 814

817 815

818 816

819 817

820 818

821 819

822 820

823 821

824 822

825 823

826 824

827 825

828 826

829 827

830 828

831 829

832 830

833 831

834 832

835 833

836 834

837 835

838 836

839 837

840 838

841 839

842 840

843 841

844 842

845 843

846 844

847 845

848 846

849 847

850 848

851 849

852 850

853 851

854 852

855 853

856 854

857 855

858 856

859 857

860 858

861 859

862 860

863 861

864 862

865 863

866 864

867 865

868 866

869 867

870 868

871 869

872 870

873 871

874 872

875 873

876 874

877 875

878 876

879 877

880 878

881 879

882 880

883 881

884 882

885 883

886 884

887 885

888 886

889 887

890 888

891 889

892 890

893 891

894 892

895 893

896 894

897 895

898 896

899 897

900 898

901 899

902 900

903 901

904 902

905 903

906 904

907 905

908 906

909 907

910 908

911 909

912 910

913 911

914 912

915 913

916 914

917 915

918 916

919 917

920 918

921 919

922 920

923 921

924 922

925 923

926 924

927 925

928 926

929 927

930 928

931 929

932 930

933 931

934 932

935 933

936 934

937 935

938 936

939 937

940 938

941 939

942 940

943 941

944 942

945 943

946 944

947 945

948 946

949 947

950 948

951 949

952 950

953 951

954 952

955 953

956 954

957 955

958 956

959 957

960 958

961 959

962 960

963 961

964 962

965 963

966 964

967 965

968 966

969 967

970 968

971 969

972 970

973 971

974 972

975 973

976 974

977 975

978 976

979 977

980 978

981 979

982 980

983 981

984 982

985 983

986 984

987 985

988 986

989 987

990 988

991 989

992 990

993 991

994 992

995 993

996 994

997 995

998 996

999 997

1000 998

1001 999

1002 1000

1003 1001

1004

4.2. Plan de Trabajo y Roles Asignados:

Acta de reunión 01:

https://docs.google.com/document/d/18sBtHKMZLHvM7OXoQjdRlZr1AlwIEXfP6kS_D3pPgJDU/edit?usp=sharing

Acta de reunión 02:

https://docs.google.com/document/d/1l6s3OzLZKTjPNMspMZn1t0_3LccOJs3Cj3zfjpUpmT0/edit?usp=sharing

Acta de reunión 03

<//docs.google.com/document/d/1q4H3kJTaohV3ng1LDI938YgBC8Mm9CUmRKzgxMQZUMI/edit?usp=sharing>

Enlace de la presentación

https://www.canva.com/design/DAGpa77Wpho/r_OBXOZWLxCX-Aw0NM8nQg/edit?utm_content=DAGpa77Wpho&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton