

| **EXAMEN PARCIAL** |
| --- |
| **RÚBRICA DE TRABAJO Y/O PROYECTO**  **Alumno: Daniel Casas** |

**EXAMEN PARCIAL**

| **Título del trabajo y/o proyecto a realizar** | **Juego de Colores** |
| --- | --- |
| **Instrucciones** | **Modalidad: Individual**   1. Revisar y leer los documentos en las carpetas Clase Desarrollada y/o Recursos de la semana 1 a la 8. 2. Lee cuidadosamente los requerimientos del examen parcial antes de responder. Este se encuentra en la siguiente liga: [EXAMEN PARCIAL.docx](https://docs.google.com/document/d/1OwxrKCHcafbDCeCqIhY1Id9OEa6VXHeU/edit?usp=sharing&ouid=116754304943457422379&rtpof=true&sd=true) 3. Utiliza el lenguaje Kotlin y Android para completar cada punto. 4. Subir el examen resuelto a un repositorio Github. Nota: Este repositorio debe ser público, debe usar este repositorio exclusivamente para este examen parcial. 5. Se valorará de acuerdo a la rúbrica adjunta. 6. Responder la rúbrica adjunta. 7. Entregar el examen parcial anexando la liga del repositorio y la rúbrica en la sección correspondiente en google classroom.   **Notas:**   * Si **dos alumnos presentan el mismo código** o existe una **sospecha de generación de IA**, se **citará al alumno para que exponga su código, resuelva preguntas, etc**; si el alumno **no puede explicar su código** fuente **tendrá cero (00)**. * Si entrega código utilizando algo **distinto** a un repositorio **Github**, ó el proyecto android se **encuentra incompleto**, tendrá **2 puntos menos.** |

| **Criterios y puntaje** | **Bueno (5 puntos)** | **Regular (3 puntos)** | **Deficiente (0-1 puntos)** |
| --- | --- | --- | --- |
| **1. Funcionamiento general del juego**  (Navegación entre fragments, lógica del juego, temporizador, feedback al usuario, ciclo de vida fragmentos, no presenta problemas al rotar pantalla) | El juego es completamente funcional. Navegación fluida entre fragments. La lógica del juego es correcta y clara.  Puntaje del alumno: 5 | El juego funciona parcialmente. Algunos errores menores o navegación confusa. | El juego no funciona correctamente o no es posible completar la interacción. |
| **2. Interfaz de usuario**  (ConstraintLayout, visibilidad, claridad, botones, retroalimentación visual) | La interfaz es clara, intuitiva y está bien organizada. Uso adecuado de ConstraintLayout. | La interfaz es comprensible pero con algunos problemas de alineación o visibilidad.  Puntaje del alumno: 3 | La interfaz es confusa, desorganizada o dificulta la interacción. |
| **3. Calidad del código**  (Nombres de variables, organización, comentarios, modularidad) | El código está limpio, bien organizado y documentado. Buen uso de funciones y estructuras.  Puntaje del alumno: 5 | El código cumple pero presenta redundancias o falta de comentarios. | El código es difícil de leer, no está organizado ni documentado. |
| **4. Funcionalidad adicional-obligatoria.**  (Ej. sonidos, animaciones, niveles, puntuación) | Se implementa correctamente al menos una funcionalidad adicional y mejora la experiencia del usuario.  Puntaje del alumno: 5 | La funcionalidad adicional existe pero está incompleta o no funciona bien. | No hay funcionalidad adicional o está mal implementada. |