

Índice

CCAR-F · Clave de respuestas

1

CCAR-F · Clave de respuestas

Cada respuesta trae el principio que la sostiene y por qué fallan las otras. El razonamiento importa más que la letra: en el examen real el escenario será distinto, el principio es el mismo.

1

Correcta: A y B

Acelerador y freno a la vez: prácticas probadas que se heredan por control de versiones, y límites explícitos sobre lo que la automatización puede ejecutar.

Por qué no las otras: C maximiza el radio de daño; D reproduce la fragmentación; E agrega rotación sin evaluación.

2

Correcta: A

Un revisor anclado al razonamiento que generó el código hereda sus puntos ciegos. Por eso la revisión se hace desde contexto fresco, separado de la sesión que escribió.

Por qué no las otras: B y C son ajustes de configuración que no tocan el anclaje; D sería un problema de proceso visible en los tiempos.

3

Correcta: A

Diagnosticar antes de intervenir. Las trazas muestran si el desvío nace en la entrada, en lo recuperado, en la descripción de la herramienta o en el prompt.

Por qué no las otras: B adivina la causa; C y D son intervenciones grandes sin evidencia de que apunten al problema.

4

#	Correcta
1	shared configuration in version control
2	review from an independent context
3	inspect the traces first
4	restrict what automation may execute
5	shared configuration in version control

Inconsistencia entre equipos y onboarding se resuelven con configuración compartida y versionada. Revisión que no encuentra nada pide contexto fresco. Conducta inesperada pide diagnóstico antes que cambio. Automatización sin supervisión pide límites de ejecución.

5

Correcta: B y C

La autonomía se justifica cuando el camino no se puede enumerar de antemano. Que las intervenciones no sigan un patrón listable y que los pasos aparezcan a mitad del proceso son exactamente esa condición.

Por qué no las otras: A y E son escala y costo: se resuelven dentro de cualquier patrón; D es prestigio, no un driver arquitectónico.

6

Correcta: A y C

El supervisor aporta secuenciación y un punto único donde queda registrado el orden de las decisiones. Eso es justamente lo que un esquema peer-to-peer no puede garantizar.

Por qué no las otras: B es trabajo del especialista; D es interno de cada agente; E es una decisión de configuración fijada de antemano, no de orquestación en ejecución.

7

Correcta: A y B

Descomponer sirve por dos motivos: cada paso recibe atención enfocada y cada salida intermedia se puede verificar. Sin salida estructurada entre pasos, la cadena vuelve a ser una caja negra.

Por qué no las otras: C confunde capacidad con estructura; D deja el prompt gigante intacto; E deshace la descomposición.

8

Correcta: A y C

Una sola llamada aumentada alcanza cuando hay una transformación y el contexto necesario cabe al lado del pedido. Es el escalón más bajo de la escalera y se elige siempre que cumpla.

Por qué no las otras: B pide multi-agente; D pide un agente; E pide un workflow con pasos discretos.

9

#	Correcta
1	the architect, at design time
2	the model, at run time
3	a human reviewer, per case
4	the architect, at design time
5	the model, at run time

Lo predecible lo fija una persona en diseño; lo que depende de lo que va apareciendo lo decide el modelo en ejecución; lo irreversible o de alto impacto lo decide un humano caso por caso.

10

Correcta: A

Cuando hay categorías de entrada claramente distintas, clasificar primero y derivar permite atender lo simple barato y rápido, y reservar el análisis profundo para lo que lo necesita.

Por qué no las otras: B paga el costo del caso complejo en cada consulta trivial; C mezcla dos tareas en un prompt; D parte la experiencia del usuario sin necesidad.

11

Correcta: A

Validar entre pasos es lo que convierte una cadena en algo depurable: el error se detecta donde nace y no contamina lo que sigue. Es la razón principal para descomponer.

Por qué no las otras: B detecta cuando ya se arrastró; C no cubre lo que el proceso genera; D confía la garantía a una instrucción.

12

Correcta: A

Una cadena en producción necesita conducta definida ante falla, no solo camino feliz. Reintento, degradación o corte explícito son decisiones de diseño que se toman antes, no en el incidente.

Por qué no las otras: B parchea aguas abajo sin saber qué falló; C baja la frecuencia pero no define qué pasa cuando ocurre; D salta un paso necesario en silencio.

13

Correcta: A

Con calidad equivalente medida, se elige el modelo más chico que alcanza el umbral y se sostiene la evaluación para confirmar que lo sigue alcanzando.

Por qué no las otras: B ignora la medición que ya existe; C hace el comportamiento impredecible sin evaluar; D baja de escalón sin evidencia de que ese modelo cumpla.

14

#	Correcta
1	few-shot examples
2	move the rule to an edge
3	prompt caching
4	retrieve only what is relevant
5	few-shot examples

Formato inconsistente se arregla mostrando, no describiendo. Una regla que se pierde es un problema de posición. Contenido idéntico repetido es caché. Un documento grande con respuestas flojas en el medio y caras es recuperación selectiva.

15

Correcta: A

Los ejemplos cuestan tokens en cada pedido. Si las definiciones ya alcanzan y la precisión es alta, agregarlos paga costo sin comprar nada. Cada técnica se aplica según beneficio medido.

Por qué no las otras: B es una regla absoluta que no se sostiene; C infla el prompt sin evidencia; D gasta capacidad donde no hay problema.

16

Correcta: A

Resumir lo viejo y conservar textual lo que decide —identificadores, importes, compromisos— combina las dos cosas: cabe en la ventana y no se pierde el dato que rompe la respuesta si falta.

Por qué no las otras: B descarta información sin criterio; C corre el problema unos turnos; D tira el hilo de la conversación.

17

Correcta: A

Las herramientas se diseñan orientadas a la tarea, no como espejo de la API interna, y solo se presentan las relevantes a cada pedido. La complejidad queda del lado del sistema, donde es determinística.

Por qué no las otras: B compra capacidad sin reducir la confusión; C agrega latencia a cada llamada; D descarta las interfaces tipadas y hace los errores más difíciles de detectar.

18

Correcta: A

El aislamiento entre inquilinos se resuelve en la consulta, no en el prompt: si el filtro por metadato no devuelve el documento, el modelo no puede citarlo aunque quiera.

Por qué no las otras: B y D dejan la separación en manos de una instrucción; C confía en que la relevancia haga de control de acceso.

19

Correcta: A

Frescura en minutos pide indexado incremental disparado por el cambio, y una reconstrucción completa periódica que corrige la deriva acumulada. Las dos piezas juntas.

Por qué no las otras: B llega con horas de retraso; C empeora deliberadamente la frescura; D pierde la búsqueda semántica.

20

Correcta: A

Las herramientas se diseñan por tarea. Espejar la API interna traslada su complejidad al modelo, que es donde peor se maneja; agrupada por tarea, la complejidad queda del lado determinístico.

Por qué no las otras: B produce exceso de capacidades; C descarta la interfaz tipada; D usa un criterio del backend para una decisión del agente.

21

Correcta: A

Se despierta a alguien por una condición sostenida que indica que el sistema está roto. Una tasa de error de herramientas por encima del umbral y mantenida es eso.

Por qué no las otras: B es ruido estadístico; C se revisa, no se despierta a nadie; D es una señal individual sin significado operativo.

22

Correcta: A

Se paraleliza el trabajo de campo, no la pluma. La voz, el hilo y el no repetirse son propiedades del documento entero, y ningún subagente las ve.

Por qué no las otras: B y D producen cuatro voces y solapes; C desperdicia el paralelismo disponible.

23

Correcta: A

En routing se descarta entre opciones que ya existían; en orchestrator se suman piezas que nacen al leer el problema, y la síntesis final es su trabajo principal.

Por qué no las otras: B elige uno de un menú fijo; C reparte secciones definidas de antemano; D es una cadena secuencial.

24

Correcta: A y B

La persona fija el contrato, el modelo llena el contenido. Sin formato común los resultados no se pueden juntar, y los límites y herramientas son decisiones de seguridad, no de contenido.

Por qué no las otras: C, D y E dependen del problema concreto y solo se pueden decidir en ejecución.

25

Correcta: A y B

Multi-agente consume del orden de quince veces más tokens que una conversación normal. Se justifica con independencia real y con subtareas que generan volumen que conviene resumir aparte.

Por qué no las otras: C y D son justamente donde fracasa: dependencia secuencial y estado compartido; E no es un criterio arquitectónico.

26

Correcta: A y B

Formatos dispares significan que falta el contrato de salida escrito por una persona. Se define una vez y se valida al recibir, antes de intentar juntar nada.

Por qué no las otras: C parchea aguas abajo; D rompe el aislamiento que justifica los subagentes; E sacrifica el paralelismo.

27

Correcta: A y B

El bloqueo no es un candado mudo: impide la ejecución y devuelve la explicación al modelo, que puede corregir el rumbo en el mismo turno.

Por qué no las otras: C es demasiado drástico; D no corre porque no hubo herramienta; E no es automático, el modelo decide con la explicación recibida.

28

Correcta: A y B

Los CLAUDE.md de subcarpeta se cargan bajo demanda, solo al tocar archivos de esa carpeta. Eso evita arrastrar las convenciones de doce aplicaciones en cada sesión.

Por qué no las otras: C es exactamente el desperdicio que se quiere evitar; D confunde configuración con instrucciones; E depende de la memoria de la persona.

29

Correcta: A

Si la salida la consume un programa, la estructura se fuerza por schema, no se pide en el prompt. Pedirla funciona casi siempre, y casi siempre no alcanza cuando hay un parser del otro lado.

Por qué no las otras: B sigue siendo una petición; C convierte el error en fallo silencioso; D parchea síntomas y acumula deuda.

30

Correcta: A

Cuando el modelo mezcla partes, el problema es que no hay frontera explícita. Delimitar con etiquetas separa instrucción, dato y pregunta sin tocar el contenido.

Por qué no las otras: B infla el prompt; C descarta material necesario; D empeora la variabilidad.

31

Correcta: A

La pregunta que ordena el dominio es quién consume la salida. Para una persona, un error de formato no rompe nada; poner un mecanismo duro donde alcanza uno blando cuesta mantenerlo y no compra nada.

Por qué no las otras: B y C sobre-corrigen; D vuelve el comportamiento impredecible.

32

Correcta: A

Los ejemplos son la especificación implícita de la tarea. Si la distribución de ejemplos no refleja las categorías reales, el modelo aprende ese desbalance.

Por qué no las otras: B y D atribuyen la falla a parámetros; C es falso.

33

Correcta: A

Entre dos y cuatro ejemplos de entrada y salida superan a cualquier descripción textual para consistencia de formato, y el modelo generaliza el patrón a casos nuevos en vez de repetir los ejemplos.

Por qué no las otras: B y C insisten en describir; D reduce variación pero no enseña la forma esperada.

34

Correcta: A

«Conservador» y «alta confianza» no son criterios. Se define exactamente qué marcar, qué no marcar y qué significa cada nivel de severidad, con un ejemplo por nivel.

Por qué no las otras: B y C atribuyen a tamaño lo que es ambigüedad; D es falso: se expresa con criterios explícitos.

35

Correcta: A

Las herramientas se diseñan orientadas a la tarea, no como espejo de la API interna. La complejidad se queda del lado del sistema, donde es determinística y testeable.

Por qué no las otras: B traslada carga al contexto; C atrapa el síntoma; D descarta la interfaz tipada.

36

Correcta: A

Es el antipatrón capital: devolver vacío ante un error es un fallo silencioso. El modelo concluye que no hay resultados y construye una respuesta segura sobre una premisa falsa.

Por qué no las otras: B es irrelevante; C esconde el problema una capa más abajo; D no describe el defecto.

37

Correcta: A

Cruzar la frontera de otra organización sin exponer lo interno es el caso que define la comunicación agente a agente: cada agente es la frontera de su propia empresa.

Por qué no las otras: B, C y D exponen justamente lo que ninguna de las dos partes está dispuesta a exponer.

38

Correcta: A

La descripción es el mecanismo de selección. Un nombre específico más el tipo concreto de dato que devuelve cada herramienta elimina la ambigüedad de raíz.

Por qué no las otras: B agrega latencia sin quitar la ambigüedad; C esconde la decisión en un parámetro; D sacrifica una capacidad que se usa.

39

Correcta: A

La configuración de proyecto se versiona con sustitución de variables de entorno: el archivo viaja al repositorio, el token no.

Por qué no las otras: B publica el secreto; C pierde el beneficio de compartir la configuración; D lo mete además en la ventana de contexto.

40

Correcta: A

El nivel de usuario vive en el directorio personal, no se comparte por control de versiones y sirve para experimentar sin tocar al equipo.

Por qué no las otras: B y D contaminan la configuración compartida; C es una rama que hay que mantener viva sin motivo.

41

Correcta: A

Un recurso que entrega el catálogo o el esquema le da al agente el mapa de entrada, y le evita gastar llamadas exploratorias para descubrir lo que ya se podía declarar.

Por qué no las otras: B no persiste entre sesiones; C describe herramientas, no datos; D invita a inventar.

42

Correcta: A

La pregunta que ordena el dominio es si la información tiene que sobrevivir sí o sí. Si la respuesta es sí, no puede vivir solamente en la conversación: se reinyecta.

Por qué no las otras: B no tiene relación; C pierde detalle y sigue dependiendo del historial; D descarta trabajo útil.

43

Correcta: A

La fiabilidad se construye con mecanismos determinísticos alrededor de un modelo que falla de forma no determinística. Sin fundamento recuperado, lo correcto es declararlo.

Por qué no las otras: B es la puerta directa a la alucinación; C traslada el problema al usuario; D repite una consulta que ya falló.

44

Correcta: A

Si un dato tiene que sobrevivir sí o sí, no puede vivir solamente en la conversación. Un bloque de hechos del caso se mantiene aparte, se actualiza y se reinyecta en cada pedido.

Por qué no las otras: B y D posponen el problema; C deja la persistencia en manos del propio modelo.

45

Correcta: A

El efecto de perderse en el medio es conocido y se mitiga por posición: lo que decide la respuesta va cerca del principio o del final.

Por qué no las otras: B duplica el costo; C no tiene relación; D destruye el orden del documento.

46

Correcta: A

Un archivo de notas sobrevive a la degradación del contexto y al cierre de sesión, y evita repetir el descubrimiento desde cero.

Por qué no las otras: B no es sostenible; C pierde justamente los detalles concretos; D desperdicia tiempo y tokens.

47

Correcta: A y B

El subagente arranca en blanco y en contexto aislado. Por eso su encargo tiene que ser auto-suficiente: objetivo, datos y formato de salida explícitos.

Por qué no las otras: C, D y E suponen una memoria o herencia compartida que no existe; ese aislamiento es justamente lo que protege el contexto del coordinador.

48

Correcta: A y B

Se abre sesión nueva cuando el contexto dejó de ser confiable: resultados viejos o deterioro tras trabajo no relacionado. Para explorar una variante desde el mismo punto de partida existe la bifurcación de sesión.

Por qué no las otras: C no justifica cortar; D pide bifurcar, no empezar de cero; E describe un contexto sano.

49

#	Correcta
1	execute the requested tools and continue
2	show the result to the user
3	raise the output budget and retry
4	treat the custom stop sequence as the boundary
5	execute the requested tools and continue

El bucle se gobierna por stop_reason: tool_use pide ejecutar y seguir, end_turn cierra, max_tokens indica truncamiento y stop_sequence marca un corte definido por vos.

50

Correcta: A y B

El hook se ejecuta en la máquina local, con los permisos del usuario y de forma automática. Por eso aceptar la configuración de un repositorio ajeno equivale a ejecutar código ajeno.

Por qué no las otras: C, D y E describen protecciones que no existen: por eso hay que revisar el contenido antes de aceptarlo.

51

Correcta: A y B

Los permisos son mecanismo duro: se declaran por herramienta con patrones y lo denegado gana siempre. El modelo no ve el archivo; se entera de que algo estaba prohibido cuando lo intenta y le rebota.

Por qué no las otras: C es falso, settings.json no entra al contexto; D contradice la naturaleza del mecanismo; E describe mal la precedencia.

52

#	Correcta
1	CLAUDE.md
2	hook
3	settings.json
4	skill or subagent
5	hook

El modelo lee CLAUDE.md; el sistema ejecuta hooks; settings.json declara el terreno sin que el modelo lo vea; skills y subagentes son procedimientos que se invocan. Pedir y garantizar son capas distintas.

53

Correcta: A y B

Las rutas relativas se resuelven respecto del archivo que importa y el anidamiento tiene un tope de cinco niveles. Sirve para partir convenciones largas en piezas mantenibles.

Por qué no las otras: C describe otra cosa; D es incorrecto, no va espacio; E es falso, se pueden importar README o package.json.

54

Correcta: A y B

El frontmatter con patrones de ruta activa cada regla solo al tocar los archivos que le corresponden. Es divulgación progresiva aplicada a las convenciones.

Por qué no las otras: C confunde petición con garantía: siguen siendo texto; D es lo contrario de lo que ocurre; E describe alcance de usuario.

55

Correcta: A y B

Compactar libera ventana comprimiendo el historial, y en esa compresión se pierden detalles concretos: cifras, fechas, nombres. Lo que se guarda como memoria vive en CLAUDE.md y se carga sola al arrancar.

Por qué no las otras: C describe algo que no ocurre; D contradice el propósito de la memoria; E no es un requisito.

56

#	Correcta
1	project-level, committed to the repository
2	user-level, in the home directory
3	a hook registered on a tool event
4	a skill with context: fork
5	project-level, committed to the repository

Lo compartido se versiona en el proyecto; lo personal vive en el directorio del usuario; lo que debe garantizarse va en un hook; lo que genera ruido se aísla en un subagente.

57

Correcta: A

Un campo obligatorio fuerza al modelo a poner algo. Solo se marca como requerido lo que siempre está disponible; el resto va opcional o nullable.

Por qué no las otras: B pide en texto lo que el esquema está exigiendo al revés; C detecta tarde y a mano; D multiplica llamadas sin resolver la causa.

58

Correcta: A

Un valor «unclear» es preferible a una categorización incorrecta, y un «other» con campo de detalle mantiene el esquema extensible sin inventar categorías.

Por qué no las otras: B persigue una cola infinita; C es lo que ya está haciendo mal; D no tiene relación con el diseño del enum.

59

Correcta: A

Extraer el valor declarado y el calculado, y marcar el conflicto, convierte una discrepancia silenciosa en un dato que alguien puede resolver.

Por qué no las otras: B y C eligen un lado y pierden la señal; D esconde el problema.

60

#	Correcta
1	Glob
2	Grep
3	Read
4	Edit
5	Glob

Glob busca por nombre y patrón de ruta, Grep busca dentro del contenido, Read carga el archivo completo y Edit modifica por coincidencia única de texto.
