Examen práctico de Analista de Desarrollo

Gerencia de Innovación

Nombre aspirante: Luis Roberto Orellana Dardón

Teléfono: 4683-2625

**Contestar las siguientes preguntas:**

1. **Bases de datos:**
   1. ¿Qué hace un left join?

La unión de 2 tablas, toma todas las filas de la tabla izquierda y los coincidentes de la derecha, basándose en una columna.

* 1. ¿Qué hace un inner join?

La unión de 2 tablas, toma las filas coincidentes basándose en una columna.

* 1. ¿Cuál es la diferencia entre el cast y convert?

El Cast es para convertir datos generales mientras que el convert te permite ser más especifico en la solicitud.

* 1. ¿Para qué sirve un índice?

Para acelerar la búsqueda de data en Base de Datos.

* 1. ¿Para qué sirve un Store Procedure?

Para predefinir un procedimiento y poder llamarlo cuando se necesite.

* 1. ¿Cuál es la diferencia entre un índice Clusteriado y no clusteriado?

El índice Clusteriado ordena físicamente en memoria los datos según ese índice mientras que el No Clusteriado no cambia ni ordena, solo crea una lista con referencias para fácil acceso.

* 1. ¿Para qué sirve una tabla temporal y como se declara?

Para guardar datos temporalmente, como cálculos. Se declara casi igual como se crea una tabla con la excepción de que lleva un “#” antes del nombre de la tabla.

* 1. ¿Para qué sirve una tabla doble temporal y como se declara?

Tiene la misma función que una tabla temporal común con el agregado de que podemos guardar automáticamente el historial de cambios.

* 1. ¿Para qué sirve la función set dateformat dmy?

Para definir un formato de fecha dia mes año.

* 1. Escriba la sintaxis de un cursor:

DECLARE @variable1 INT

DECLARE @variable2 VARCHAR(100)

DECLARE cursorEmpleado CURSOR FOR

SELECT id, nombre FROM empleado

OPEN cursorEmpleado

FETCH NEXT FROM cursorEmpleado INTO @variable1, @variable2

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT 'ID: ' + CAST(@variable1 AS VARCHAR) + ', Nombre: ' + @variable2

FETCH NEXT FROM cursorEmpleado INTO @variable1, @variable2

END

CLOSE cursorEmpleado

DEALLOCATE cursorEmpleado

* 1. ¿Diferencia entre un truncate table y un drop table?

Truncate elimina todos los datos de una tabla, mientras que el drop elimina la tabla por completo.

* 1. ¿Indique una herramienta para ver el performance de un gestor de base de datos y que encontramos?

Activity Monitor dentro de SSMG, desde ahí podemos ver el performance de la base de datos en términos generales y específicos. Como procesos activos, CPU y Memoria en uso, etc.

2 **Backend:**

2.1 Explica cómo implementarías una estrategia de cacheo a nivel de aplicación sin utilizar servicios externos (Se puede apoyar de una base de datos relacional como Postgresql o sql server al ser un escenario común donde no se cuenta con la disponibilidad de usar servicios de alguna base de datos dictionary in memory como Redis):

En lo personal utilizaría una tabla en SQL llamada cache donde aloje todos los datos cacheados y mi idea de implementación sería:  
Cada que el usuario realice una petición se busca en la tabla y verifica si el datos existe y aún no ha sido expirado, sí en dado caso el dato ya expiro o no existe ya realiza la consulta por un dato real y obviamente lo guardamos en la tabla cache con fecha y hora. Adicional crearía un job para limpiar la data expirada así evitando lentitud y saturar la memoria.

2.2 Explica el concepto de versionado de APIs y cómo lo implementarías.

Este se utiliza para controlar la versión de las APIs, al realizar cambios o mejorar en las mismas, yo lo interpretaría agregar un V1, V2, en la url de cada una.

2.3 ¿Qué son los endpoints idempotentes y por qué son importantes?

Básicamente es una petición que sin importar cuantas veces lo llames te dará el mismo resultado y su importancia influye con la seguridad del sistema, lo vuelve más seguro y óptimo.

2.4 ¿Qué es la inyección SQL y cómo prevenirla?

Es cuando un usuario malicioso encuentra una irregularidad por ejemplo en un login o registro que solicita un campo texto en el cual insertar una consulta SQL para realizar un insert masivo de data así afectando el funcionamiento de esta y yo la prevengo utilizando Store Procedure en el caso de logins.

2.5 Explica cómo implementar autenticación y autorización en una aplicación backend (incluyendo como manejarías el almacenamiento seguro de contraseñas) para evaluar su conocimiento de encriptación simétrica y asimétrica.

Autenticación: Cliente ingresa usuario y contraseña (la contraseña esta en criptada, nunca en texto plano) realiza un request a la base de datos y si es válida envía un Token que contiene la información del usuario con firma de autenticación.

Autorización: Para cada endpoint se solicita un token el cual se verifica recibido en los headers y luego por medio de Roles y Accesos se brinda el acceso a la data que solicite el endpoint.

2.6 ¿Cómo identificarías y resolverías un cuello de botella en una aplicación backend?

Lo identificaría desde el Activity Monitor del gestor de BD, identifico que proceso es el que genera el problema y lo atacaría de la siguiente forma:  
 1. Implementación de índices.

1. Paginación de datos.
2. Optimización de querys.

2.7Explica las diferencias entre concurrencia y paralelismo y cómo afectan el diseño de aplicaciones backend.

Concurrencia son tareas que se ponen en cola y ejecuta una a la vez, este afecta los recursos asì que hay que definir los recursos compartidos. Mientras que el paralelismo las ejecuta al mismo tiempo este afecta la memoria y CPU por la ejecución de varios procesos al mismo tiempo.

2.8 ¿Cómo manejarías la sincronización de hilos en un entorno multihilo para evitar condiciones de carrera?

Utilizaria semáforos para indicar cuando si y no puede un hilo acceder o estructura de colas.

2.9 ¿Cuáles son los problemas asociados con el uso de números de punto flotante en aplicaciones financieras? Proporcione ejemplos específicos.

Que suelen aproximar automáticamente la cantidad de decimales, por ejemplo al realizar un calculo de interés normal o moratorios y utilizar tasas con màs de 2 decimales, por lo que en mi experiencia he utilizado tipo de datos numeric para SQL y me ha funcionado muy bien.

2.10 ¿Cómo afecta el uso excesivo de bloques try-catch y la implementación de recursión profunda al rendimiento de una aplicación backend? En tu respuesta, analiza el impacto en el uso de la CPU y la memoria RAM, y explica los conceptos de ciencias de la computación que sustentan estos efectos.

Este puede tener un consumo excesivo de CPU si es usado donde no se debe y también memoria por el trazado de objetos. En la Memoria puede causar un desbordamiento de pila por la memoria ocupada en cada stack si es que existe recursión excesiva, y este proceso también consume CPU.

1. **Desarrollo web y mobile**
   1. ¿Cuáles son las principales diferencias entre el Renderizado del Lado del Servidor (SSR) y el Renderizado del Lado del Cliente (CSR)? Analiza sus ventajas y desventajas en términos de rendimiento, SEO y experiencia de usuario.

La mayor diferencia es que SSR por ejecutarse en el servidor consume recursos mientras que el CSR se ejecuta en cada navegador del cliente por lo que nos ahorra recursos, el detalle es que eso vuelve un poco lento la primera ejecución de la página pero fluida ya que carga pero si queremos aumentar la experiencia de usuario me inclinaría por el CSR ya que tiene mayor fluides ya cargada la página.

* 1. Explica qué es el Critical Rendering Path en el contexto del desarrollo frontend y describe las técnicas avanzadas que utilizarías para optimizarlo, reduciendo el tiempo de carga percibido por el usuario.

El termino se refiere a la secuencia para construir una pagina web dentro de un navegador. Para optimizar esto cargarìa el CSS solo cuando es requerido en el contenido visible de igual manera haría lo mismo con las imágenes y como punto final reduciría las solicitudes HTTP.

* 1. Describe cómo implementarías un middleware personalizado en cualquier gestor de estados (Redux o Zustand) para manejar efectos secundarios complejos, como llamadas a APIs externas y gestión de autenticación, asegurando que el flujo de datos permanezca predecible y mantenible.
  2. Explica las diferencias entre REST y GraphQL en el contexto del desarrollo frontend y describe cómo implementarías una solución híbrida que aproveche las ventajas de ambos enfoques.

El modelo REST te permite obtener datos con el detalle que a través de URL fijas mientras que el GraphQL es màs flexible en cada petición.

Utilizarìa el REST para consultas básicas como un listado de usuarios, su ventaja es que será mucho màs rápido y eficiente. Utilizarìa GraphQl para consultas màs especificas como detalle de cada usuario y filtros agregados, su beneficio es la flexibilidad.

* 1. Explica los conceptos de Progressive Enhancement y Graceful Degradation en el desarrollo frontend y cómo los aplicarías para asegurar una experiencia de usuario consistente en diferentes navegadores y dispositivos.

Progressive Enhancement: se enfoca en partir desde lo básico hacìa lo avanzado mientras que el Graceful Degradation al revés, desde lo complejo y se va adaptando a lo básico

Con Progressive Enhancement haría una web puede mostrar texto e imágenes básicas sin JavaScript, y luego cargar animaciones y formularios avanzados solo en navegadores modernos. Y con Graceful Degradation un sitio con efectos CSS y animaciones puede mostrar una versión estática o simplificada en navegadores antiguos que no soportan esas tecnologías.

* 1. Explica la importancia de la accesibilidad en el desarrollo y describe las mejores prácticas para asegurar que una aplicación web o mobile sea accesible para todos los usuarios, incluyendo aquellos con discapacidades o poca educación digital (usuarios de escasos recursos con dispositivos demasiado antiguos y con los que les cuesta interactuar).

Es super importante para ser inclusivo y procurar que cualquier persona sin importar su capacidad pueda ingresar a un sistema, de las mejores practicas considero que utilizar adecuadamente el HTML con las etiquetas correctas para que los navegadores interpreten bien su función, un buen UX/UI, permitir que la app funcione únicamente con teclado, diseño responsivo y ligero en tema de recursos.

* 1. Describe las Core Web Vitals (LCP, FID y CLS) y explica cómo optimizarías una aplicación frontend para cumplir con estos indicadores de rendimiento.

Estas con métricas creadas por Google que califican los puntos importantes de una web, para atacar LCP optimizarìa la fluidez, usando lazy loading para contenido visual, comprimir contenido tipo imágenes, evitar scripts innecesarios.

Para atacar FID utilizarìa de forma optima JS evitando renders innecesarios y tareas largas.

Para atacar CLS reservaría espacio único para archivos visuales(imagenes) y utilizar la menor cantidad de layout con animaciones.

* 1. En el contexto de desarrollar un chat frontend que interactúa con una inteligencia artificial (IA) a través de una API de un modelo de lenguaje grande (LLM), ¿qué tecnología utilizarías para manejar la comunicación en tiempo real: WebSockets o Server-Sent Events (SSE)? Detalla los beneficios de la tecnología seleccionada y explica cómo gestionarías la conexión, el envío y la recepción de mensajes

Para un chat con IA utilizarìa WebSockets porque permite conexión y consultas en tiempo real y es escalable para multiples chats o tareas.

1. **Inteligencia Artificial**

4.1 ¿Cuáles son los principales algoritmos de aprendizaje superficial (shallow learning) y en qué casos usarías cada uno?

Regresión Lineal para la predicción de datos numericos, como precios, ganancias estimadas, mora, etc.

Árboles de decisión, para como lo dice la palabra tomar decisiones basadas en hechos.

4.2 ¿Qué es la regresión lineal y cuándo la emplearías?

es una técnica utilizada para formar una estadística lineal a partir de datos presuntamente sin patrones. La utilizaría para predicción de datos numéricos como % de morosidad en cartera cantidad de pagos recibidos, etc.

4.3 ¿Cómo funciona un árbol de decisión y cuál es la diferencia entre criterio Gini y entropía?

este funciona representando posibles resultados y decisiones utilizando nodos.

4.4 ¿En qué se diferencian un SVM con kernel lineal y uno con kernel RBF?

Desconozco

4.5 ¿Qué efecto tiene limitar la profundidad de un árbol de decisión?

No crear un bucle infinito de toma de decisiones y volverlo más simplificado y veloz.

4.6 ¿En qué se diferencian bagging y boosting, y cómo reduce cada uno el error del ensemble?

Desconozco

4.7 ¿Cómo elegirías el número de clusters en K-Means usando el método del codo y la silueta?

Desconozco

4.8 Define la matriz de confusión y explica cómo se relacionan precisión, recall y la probabilidad

posterior de Bayes.

Desconozco

4.9 ¿En qué consiste un modelo generativo estocástico como Naive Bayes y cómo difiere de un enfoque discriminativo?

Desconozco

4.10 ¿Cuál es el propósito de la regularización L1 vs L2 y cómo afecta a los coeficientes del modelo?

Desconozco

4.11 ¿Qué arquitecturas de redes (CNN, RNN, Transformer) se usan típicamente en visión, sonido y NLP, y qué ventajas ofrece cada una?

CNN para vision, muy buena con reconocimiento fácil. Son muy rápidaz.

RNN/GRU para audio y sonido, super para reconocimiento de voz, su ventaja es que procesa toda la secuencia de un solo.

Transformer/GPT pueden ser Bidireccionales o autoregresivas, su ventaja es poder capturar relación entre palabras sin importar la longitud del texto.

4.12 ¿Cuál es el propósito de una función de activación y por qué ReLU es tan popular?

Desconozco

4.13 ¿Cómo actúa el dropout para prevenir el sobreajuste en redes profundas?

Ayuda a regar el aprendizaje con las demás neuronas para prevenir perdida de información y no depender de una especificamente.

4.14 ¿En qué se diferencian las convoluciones 1D, 2D y 3D y sus aplicaciones típicas?

Desconozco

Desconzco

4.15 Explica la atención escalada en Transformers y por qué su coste es O(n²).

4.16 ¿Podrías explicar detalladamente cómo se estructuran y se interrelacionan los procesos de propagación hacia adelante (forward propagation) y propagación hacia atrás (backpropagation) durante el entrenamiento de una red neuronal utilizando el algoritmo de descenso de gradiente? En tu explicación, incluye:

¿Qué es el descenso de gradiente y por qué es fundamental en el entrenamiento de redes neuronales?

Desconzco

4.17 ¿Qué papel juegan las operaciones de convolución en los modelos de visión por computador, y cómo se comparan con enfoques modernos como Vision Transformers? Además, ¿cuáles son las ventajas y limitaciones de utilizar servicios en la nube como AWS Rekognition para tareas de visión frente a modelos personalizados?

Su principal funcion es detectar patrones visuales funcionan aplicando filtros que se repartes tareas, uno verifica bordes otro esquinas, etc. En cambio los VIT tienen un enfoque más moderno y parten la imagen en pixeles no utiliza filtros. las ventajas de utilizarlos en AWS es que suele ser más fácil la implementación de este pero el costo suele ser mayor.

4.18 ¿Qué técnicas ofrece OpenCV para mejorar imágenes degradadas por ruido o baja iluminación antes de aplicar detección de bordes o segmentación, y cómo se comparan con enfoques basados en redes neuronales?

Desconozco

4.19 ¿Qué es la calibración de cámara en OpenCV, por qué es importante en visión estereoscópica o reconstrucción 3D, y cómo se realiza usando un patrón de tablero de ajedrez?

Se refiere a la calibración de angulos y distancia del objeto a modelar utiliza un tablero de ajedrez para utilizar los cuadros como un plano cartesiano y así realizar una estimación de distribución de tamaño de la pieza.

4.20 ¿Cómo se puede utilizar visión por computador para estimar el estilo artístico de una imagen (por ejemplo, impresionismo, cubismo, arte pop), y qué técnicas como “Gram Matrix” o redes tipo VGG se utilizan en el proceso de “style transfer”? ¿Qué limitaciones tienen estos enfoques y qué avances modernos existen para lograr una transferencia de estilo más realista y eficiente?

Primero recolectando información y caracteristicas básicas de cada stilo y alimentar el modelo AI con esto, luego aplicar filtros para determinar a que caracteristicas apunta más.

4.21 ¿Qué retos presenta la implementación de visión por computador en dispositivos móviles o embarcados (como Raspberry Pi o Jetson Nano), y cómo se puede optimizar el uso de OpenCV para estos casos?

Primero que nada la compatibilidad con los diferentes sistemas operativos del sistema y limitación de recursos ya que estos modelos requieren de bastante memoria ram y cpu. OpenCV nos puede ayudar reduciendo la resolución para tener un menor consumo de recursos.

4.22 ¿Cómo influye el diseño de prompts (zero-shot, few-shot, chain-of-thought, tool-aware) en el comportamiento y capacidades de un LLM, y qué implicaciones tiene esto para la planificación, la interpretación de instrucciones complejas y la interacción con herramientas externas?

Desconozco

4.23 ¿Qué define a un “agente” en el contexto de la IA y en qué se diferencia de un simple modelo de ML?

Un agente piensa, analiza y acciona, mientra que un modelo ML es una funcion entrenada para hacer algo sin analizar, solo ejecutar.

4.24 ¿Qué impacto tiene el uso de etiquetas XML en los prompts sobre el comportamiento y la interpretación del modelo?

Desconozco

4.25 Diseña un prompt largo “sandwich” (goal–persona–context–goal) para un LLM que genere un plan de proyecto de software, explicando cada capa.

Quiero que generes un plan detallado de desarrollo de software dividido por sprints, usando buenas prácticas de gestión ágil.

Imagina que eres un ingeniero senior de software con experiencia liderando proyectos web complejos, y estás ayudando a un pequeño equipo de desarrolladores a estructurar su trabajo.

El proyecto consiste en crear una plataforma web para gestionar viajes de transporte pesado. Se usará Laravel para el backend, React para el frontend y MySQL como base de datos. El equipo consta de 4 personas: 2 backend, 1 frontend y 1 QA. El proyecto debe completarse en 3 meses (6 sprints de 2 semanas). Debes incluir funcionalidades como gestión de usuarios, vehículos, viajes, pagos y reportes. El equipo utilizará Scrum.

Genera un plan dividido por sprints, indicando en cada uno las tareas principales, las funcionalidades clave, los objetivos de QA y cualquier riesgo o recomendación.

4.26 ¿Cómo implementaría la autenticación y el control de acceso en un servidor MCP para proteger datos sensibles?

Implementando Autenticación por token JWT, autorización(roles y accesos), encriptación de contraseñas y el uso correcto de headers.

4.27 ¿Qué técnicas de razonamiento puedes incorporar en un agente para que no solo actúe, sino que justifique sus decisiones (ej. CoT, ToT, SoT)?

Desconozco

4.28 ¿Cuáles son los componentes básicos de un agente (perfil, memoria, razonamiento, herramientas, planificación y evaluación)?

Perfil,Es como el agente captura la información. Memoria, es donde y en que estructura guarda esos datos. razonamiento, es que algoritmo utilizará para ejecutar lo solicitado. herramientas, a que servicios se conecta para extraer la data o realizar el proceso solicitado. planificación, Poner en cola los procesos. Evaluación, monitoreo de la respuesta dada al usuario.

4.29 ¿Cómo funciona la arquitectura de un agente autónomo frente a un agente asistente o proxy?

Un agente autonomo puede autogestionarse sin necesidad de un humano y el asistente funciona como un intermediario entre un modelo de datos y un usuario.

4.30 ¿Cómo diseñarías un sistema multiagente en el que diferentes agentes con capacidades especializadas (como planificación, ejecución, búsqueda o memoria) colaboran para resolver una tarea compleja de forma autónoma? Describe cómo se comunican, se coordinan y qué arquitectura facilitaría su escalabilidad y robustez.

4.31 ¿Qué patrones de coordinación (broadcast, pipeline, blackboard) existen entre agentes y cuándo usar cada uno?

Broadcast lo utilizaría cuando tengo que enviar una misma información a varios agentes, tipo alertas.

Pipeline lo utilizaría para encadenamiento, ejemplo cada uno realiza una etapa de un flujo.

Blackboard para el trabajo en conjunto.

4.32 ¿Cómo integras feedback humano en bucles de control para mejorar las decisiones de los agentes?

En la etapa final de evaluación y medir la tasa de efectividad de respuesta del modelo y corregirlo en dado caso se necesite.

4.33 ¿Cómo evalúas el rendimiento y la efectividad de un sistema multi‑agente (KPIs, trazabilidad de decisiones)?

Lo haría con KPI’s para segmentar cada etapa y su punteo, determinando donde es donde se debe corregir y atacar ese problema.

4.34 Imagina que estás diseñando un sistema de inteligencia artificial para coordinar la respuesta a emergencias en una ciudad inteligente. ¿Cómo decidirías entre implementar

Esta cortada la pregunta.

4.35 ¿Qué papel juegan los “agentic behavior trees” o los “policy planners” en la toma de decisiones de agentes complejos?

Los behavior trees permiten definir comportamientos jerárquicos y modulares. Los policy planners, generan planes para obtener objetivos considerando diferentes escenarios y estados posibles

4.36 ¿Qué criterios considerarías al elegir un framework para construir un sistema de agentes inteligentes (p. ej., autónomos, colaborativos, multimodales), y cuál elegirías entre opciones como TEN, Pipecat, LangGraph, Agno, Atomic Agents o Arklex, y por qué?

Desconozco

4.37 ¿Qué ventajas ofrece el uso de structured outputs (por ejemplo, respuesta en JSON o Pydantic) en la interacción entre agentes o con herramientas externas?

Controlar a detalle las salidas y poder obtener datos de ellas automaticamente.

4.38 ¿Cómo implementarías un agente basado en modelos de lenguaje (LLM) que mejore su rendimiento mediante aprendizaje por refuerzo utilizando herramientas como Agent Reinforcement Trainer (ART)? Describe cómo integrarías componentes clave como la asignación de recompensas, la gestión de trayectorias y la actualización de modelos.

4.39 ¿Cómo garantizarías la reproducibilidad y auditabilidad de las decisiones de un agente que interactúa con APIs o herramientas externas?

Yo registraría cada interacción del agente con APIs o herramientas externas, como entradas, salidas, marcas de tiempo y contexto.

4.40 ¿Qué es un tokenizer en un LLM y por qué es importante entender cómo tokeniza tu entrada y salida?

Es la herramienta que divide el texto en unidades más pequeñas llamadas tokens, es importante porque afecta cómo el modelo procesa y genera texto, influye en la longitud máxima de entrada, el costo computacional y la precisión en la interpretación.

4.41 ¿Qué estrategias puedes usar para reducir el número de tokens en un agente sin perder semántica?

Podemos usar técnicas como resumir o condensar el texto, eliminar palabras innecesarias o redundantes, usar sinónimos más cortos, y simplificar frases manteniendo la esencia.

4.42 ¿Cuáles son los beneficios de incorporar un sistema de memoria persistente en agentes de inteligencia artificial, y cómo impacta en su capacidad para mantener contexto, personalizar respuestas y adaptarse a cambios en el tiempo? ¿Qué desafíos surgen al manejar hechos dinámicos, múltiples sesiones y datos empresariales en evolución, y cómo se pueden abordar con arquitecturas modernas de memoria? (ejemplo: Zep AI Agent Memory)

Siempre se mantendrá en contexto ya que incorpora memoria y siempre sigue aprendiendo y haciendo un record de datos.

4.43 Qué ventajas aporta el Model Context Protocol (MCP) frente a integraciones con function calling para los LLM?

La seguridad de la información, vuelve mucho más seguras las conexiones.

4.44 ¿Cómo diseñarías un agente que lea documentos, extraiga imágenes relevantes y devuelva los nombres de los archivos junto con su contexto textual?

Utilizaria la tecnologia PyMuPDF, para leer los documentos a partir de eso utilizo parametros predefinidos de caracteristicas comunes para extraer lo solicitado.

4.45 ¿Qué técnicas de caching y persistencia usarías para evitar reprocesar archivos ya vectorizados en agentes que consultan grandes volúmenes de datos?

Utilizaría hash de contenido para asignarle una clave unica a cada contenido y así fácilitar su busqueda.

4.46 ¿Cómo estructurarías y almacenarías documentos para ser usados en un sistema RAG?

Los almacenaria en buckets donde una tecnología los lea, extraiga la data plana(texto) y los deseche para no incurrir en memoria innecesaria.

4.47 ¿Cómo combinarías búsqueda semántica con clasificación o reranking para mejorar resultados de RAG?.

Desconozco

4.48 ¿Qué es un Deep Research Agent y en qué se diferencia de otros tipos de agentes de búsqueda?

Este como la palabra lo indica realiza una busqueda profunda de datos, y brinda datos más especificos que otros agentes tipo GPT.

4.49 ¿Qué es un Computer Use Agent y en qué se diferencia de otros tipos de agentes de búsqueda?

Es un modelo que se utiliza para simular al humano y utilizar interfaces visuales de software, el automatiza procesos manuales como el registro de un formulario mientras que un agente normal, busca información y la expresa.

4.50 ¿Qué ventajas le ves a utilizar una librería como Instructor en lugar de hacer una llamada directa a una API de un LLM que devuelve respuestas en formato JSON definido?

En la segmentación de información y mejorar el input así mejorando la respuesta.

4.51 Explica qué es computación evolutiva

Desconozco

1. **Prueba Practica React, .Net y Flutter (Opcional: Kotlin o Swift)**

**2.1 UNICAMENTE REACT y .NET (MVC o MINIMAL APIS)**

Para esta **Prueba Práctica de React, .Net y Flutter** (con la opción de emplear Kotlin o Swift en lugar de Flutter) se te encarga crear un sistema integral para La Cazuela Chapina, negocio que combina dos pilares de la cocina guatemalteca: los tamales tradicionales y las bebidas de maíz y cacao.

Los tamales se venden por unidad, media docena o docena, y cada pieza puede personalizarse según cuatro atributos: tipo de masa (maíz amarillo, maíz blanco o arroz), relleno (recado rojo de cerdo, negro de pollo, chipilín vegetariano o mezcla estilo chuchito), envoltura (hoja de plátano o tusa de maíz) y nivel de picante (sin chile, suave o chapín).

Las bebidas artesanales se ofrecen en vasos de 12 oz y jarros de 1 litro. El cliente elige entre atol de elote, atole shuco, pinol o cacao batido; decide si endulza con panela, miel o prefiere sin azúcar; y, si lo desea, añade malvaviscos, canela o ralladura de cacao como topping.

Para aumentar el valor del ticket hay tres combos predefinidos. El Combo Familiar “Fiesta Patronal” incluye una docena surtida de tamales y dos jarros familiares; el Combo Eventos “Madrugada del 24” suma tres docenas, cuatro jarros y un termo de barro conmemorativo; y un Combo Estacional varía según la época – fiambre en noviembre, quema del Diablo en diciembre y recetas de Cuaresma en Semana Santa –, por lo que la aplicación debe permitir crear o ajustar combos sin redeploy.

Detrás del mostrador se controla inventario de materias primas (masa, hojas, proteínas, granos, endulzantes, especias), empaques y combustible, registrando entradas, salidas, costos y mermas. Un panel de inicio presenta indicadores clave: ventas diarias y mensuales, tamales más vendidos, bebidas preferidas por franja horaria, proporción de picante vs no picante, utilidades por línea y desperdicio de materia prima.

La versión móvil replica la funcionalidad, añade la gestión de sucursales con sus reportes y permite registrar ventas fuera de línea. Cada venta o fin de cocción emite una notificación push. Se valoran módulos opcionales como telemetría de vaporeras, cronometraje de lotes, gestión de proveedores, mapa de mesas y logística de envíos. Para la interfaz web se otorgan puntos extra si grabas un análisis con React Scan y explicas las mejoras sugeridas, sin necesidad de implementarlas.

### Guía ordenada de requisitos

1. Tecnologías objetivo  
   1.1 Frontend web en React  
   1.2 API en .Net  
   1.3 App móvil en Flutter (opcionalmente Kotlin o Swift) con el mismo backend
2. Catálogo de productos  
   2.1 Tamales: unidad, 6 uds, 12 uds  
   2.2 Atributos de tamal: masa, relleno, envoltura, picante  
   2.3 Bebidas: vaso 12 oz, jarro 1 L  
   2.4 Atributos de bebida: tipo, endulzante, topping
3. Combos  
   3.1 Combo Familiar “Fiesta Patronal”  
   3.2 Combo Eventos “Madrugada del 24”  
   3.3 Combo Estacional editable en producción
4. Inventario  
   4.1 Materias primas (masa, hojas, proteínas, granos, endulzantes, especias)  
   4.2 Empaques y combustible  
   4.3 Entradas, salidas, mermas, costos promedio  
   4.4 Bloqueo de venta por punto crítico
5. Dashboard mínimo  
   5.1 Ventas diarias y mensuales  
   5.2 Tamales más vendidos  
   5.3 Bebidas preferidas por horario  
   5.4 Proporción picante vs no picante  
   5.5 Utilidades por línea  
   5.6 Desperdicio de materias primas
6. Integración LLM obligatoria  
   6.1 API OpenRouter (Modelos gratuitos)  
   6.2 Implementación libre, se calificara la creatividad  
   6.3 Puntos extra: voz a texto, MCP de demanda
7. Funciones opcionales de alto impacto  
   7.1 Cronometraje de lotes  
   7.2 Gestión de proveedores  
   7.3 Plano de mesas  
   7.4 Logística de envíos a domicilio
8. Web – puntos extra  
   8.1 Video corto con React Scan   
   8.2 Análisis de resultados  
   8.3 Propuesta de mejoras (conceptual) no es necesario implementar
9. exclusivo  
   9.1 Gestión y reportes de sucursales  
   9.2 Registro de ventas en dispositivo  
   9.3 Notificaciones push por venta y fin de cocción

**IMPORTANTE: se estará calificando diseño, seguridad, rendimiento.**

**FECHA MÁXIMA DE ENTREGA: MARTES 5 DE AGOSTO A LA 11 PM**

Enviar un video explicando el proyecto (no hay necesidad de desplegar el proyecto), documentación y adjuntar el o los repositorios

de su proyecto a los siguientes correos

[esantos@genesisempresarial.com](mailto:esantos@genesisempresarial.com)

[cfernandez@genesisempresarial.com](mailto:cfernandez@genesisempresarial.com)

[tdeleon@genesisempresarial.com](mailto:tdeleon@genesisempresarial.com)

jmolina@genesisempresarial.com

aagonzalez@genesisempresarial.com

**DUDAS WHATSAPP: 33568980 o 30816121**