

Arquitectura (30)			Modelo de Dominio (35)					Persistencia (30)		Final
1 (10)	2 (10)	3 (10)	1 (20)	2 (5)	3 (5)	4 (5)	5 (5)	1 (20)	2 (10)	

- Se aprueba con **6** puntos y 50 % de cada punto
- Entregue cada punto en hojas **separadas**, escriba en una sola carilla, ponga apellido y **numere cada hoja**.
- **Lea todo el examen antes de comenzar**

RebaH

RebaH (reclutamiento en base a habilidades) es una plataforma en línea para la evaluación y mejora de habilidades en programación y tecnología, centrada en la resolución de desafíos y ejercicios prácticos. A través de RebaH las empresas pueden contactarse con profesionales y evaluar sus habilidades.

Preguntas

RebaH cuenta con una gran base de preguntas. Cada una está formada por una consigna, una o más habilidades (algoritmos, arquitectura, ia, bases de datos, etc...) y una colección de distintas respuestas correctas. Además, existen dos tipos tipos de preguntas: común y de código. Cuando un candidato resuelve una pregunta, se recolectan para ésta **métricas**, como por ejemplo cuánto tiempo le llevó responderla, cuánto tiempo estuvo en otra pestaña, que patrón de movimiento del mouse tuvo, etc. Como no se tiene información detallada del formato y cómputo de estas métricas, pueden considerarse de "caja negra", las mismas quedan fuera del alcance de nuestro problema.

Candidatos y evaluaciones

Los candidatos son personas que quieren ser examinadas por RebaH y se anotan a distintas evaluaciones, que consisten en un conjunto de preguntas. Los resultados de tales evaluaciones son almacenados para su posterior consideración en campañas de reclutamiento.

Campañas de reclutamiento

Las empresas, cuando necesitan contratar a una persona, pueden iniciar una campaña de reclutamiento. Para eso, eligen una evaluación de cualquier tipo como referencia para evaluar a los candidatos que la hayan resuelto previamente. Los candidatos serán ordenados por una función de ponderación (que será explicada más adelante), para luego ser contactados en ese orden.

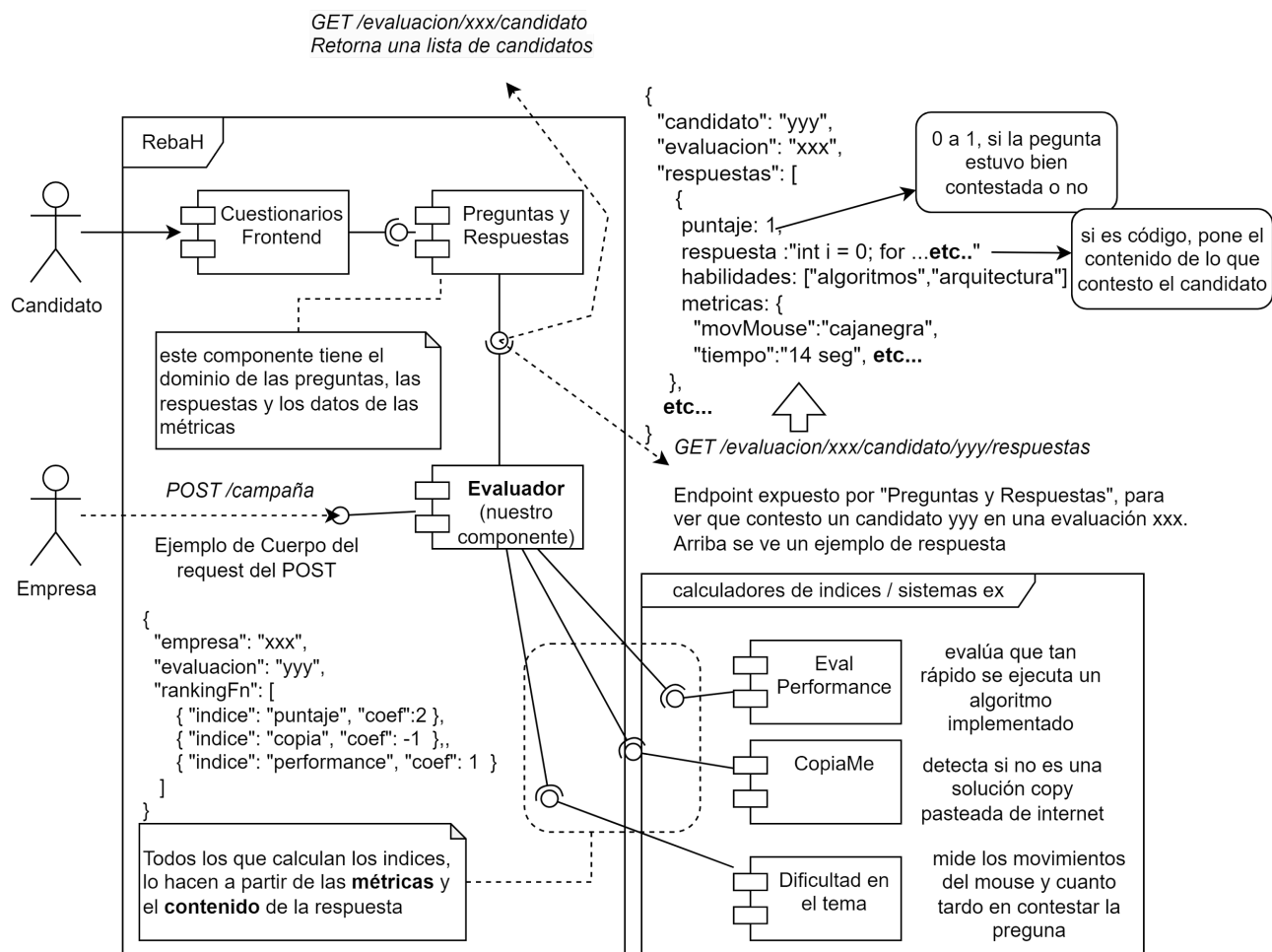
Índices complementarios

Como el puntaje obtenido en la evaluación por sí mismo es engañoso o incompleto para evaluar a un candidato, no sólo se tendrán en cuenta los puntajes obtenidos (de cada respuesta 0 o 1, según si es incorrecta o correcta) sino también que se consideran **índices**. Los mismos son valores numéricos entre 0 y 1, que se calculan en base al **contenido** y las **métricas** (una o más métricas por índice) **de cada respuesta**. Estos índices, en general, son calculados por componentes externos. Cada índice debe saber qué métricas necesita para calcularse.

Función de ponderación

Para ordenar a los participantes dentro de una campaña de reclutamiento se aplicará una función de ponderación, definida por la empresa. Esta se construirá indicando qué parámetros son necesarios (entre los que se pueden elegir el puntaje de la respuesta y los resultados de cada índice) y su importancia a la hora de evaluar un candidato (expresada mediante coeficientes numéricos).

Ejemplo completo



función de ranking $zzz = \sum_{\text{preguntas}} 2 * \text{puntaje}_i - 1 * \text{copia}_i + 1 * \text{performance}_i$

Esta función suma, por pregunta, el puntaje obtenido, el índice de copia y el de performance, cada uno con sus respectivos coeficientes (2,-1 y 1), y suma ese resultado para todas las preguntas

Candidato	Pregunta de evaluación yyy	puntaje	indice copia	indice rendimiento
A	1	1	0	0
A	2	0	1	0
B	1	1	0	1
B	2	1	0	0

Resultado de la campaña

Candidato	Puntaje final	Explicación
B	5	$((2*1) + (-1*0) + (1*1)) + ((2*1) + (-1*0) + (1*0))$
A	1	$((2*1) + (-1*0) + (1*0)) + ((2*0) + (-1*1) + (1*0))$

Consideraciones importantes

- Nuestro trabajo es resolver el componente **evaluador**, el cual se encarga de procesar las campañas.
- Debemos asumir que las preguntas ya fueron contestadas y evaluadas y sus métricas fueron capturadas: todo eso es responsabilidad del componente de "Preguntas y Respuestas". Este componente, no obstante, no conoce sobre los índices, los cuales forman parte del dominio del **evaluador**.
- A futuro se deben poder agregar nuevos **índices**
- Para ahorrar recursos, los índices sólo se calcularán en el momento que lo requiera la campaña, y solo los necesarios para calcular el ranking.
- Es importante guardar el resultado del reclutamiento y los datos necesarios para entender cómo se llegó al puntaje final.
- **No** es lo mismo una métrica que un índice, este último se calcula a partir de los primeros
- Los componentes externos que calculan los índices tienen interfaces diferentes y pueden cambiar frecuentemente, pero todos toman como entrada una o más de las métricas de las respuestas y su contenido.
- El posterior contacto entre la organización que recluta y el candidato queda fuera del alcance

Dominio (40)

Diseñar el modelo de dominio del sistema que resuelva los requerimientos aplicando el paradigma OO.

1. Armar la especificación del modelo utilizando diagramas UML (diagrama de clases obligatorio) (20p)
2. Justificar las decisiones de diseño que se tomen, por ejemplo, haciendo referencia a los principios que guían al diseño o las consecuencias de aplicar un determinado patrón. (5p)
3. Explique y dé un ejemplo de cómo se agregaría un nuevo índice (5p)
4. Mostrar sobre un diagrama de objetos o esquema conceptual, como se modela una campaña, ANTES de calcular los índices/ranking (5p)
5. Explique con un diagrama de secuencia o pseudocódigo, cómo el escenario anterior genera el resultado de la campaña (listado de candidatos ordenados ordenado por ranking). (5p)

Arquitectura (30)

1. Indicar 2 mecanismos de integración para calcular los índices a través de los sistemas externos, compárelos respecto a eficiencia y tolerancia a fallos.

2. Se observó que hay picos de reclutamiento de pocos días en determinados momentos, pero en el resto del año el sistema suele estar "tranquilo". Durante esos picos se experimentan problemas y se saturan los sistemas, en particular la CPU por el procesamiento de los ranking en el Evaluador. (a) Dé una posible solución y represéntela en un diagrama de despliegue (b) Explique qué mecanismos / conceptos / herramientas se utilizan y cómo resuelven el problema.

3. El evaluador actualmente expone una API a las empresas/organizaciones, pero se está pensando en ofrecer una interfaz gráfica. Suponemos que será utilizada mayormente desde un navegador web en una PC. Proponga 2 alternativas de implementación y compárelas según facilidad de mantenimiento y usabilidad.

Persistencia (30)

Diseñar el modelo de datos del componente al sistema para poder persistir en una base de datos relacional a través de un ORM.

1. Armar la especificación usando un DER físico. Indicando las entidades, sus campos, claves primarias, las foráneas, cardinalidad, modalidad y las restricciones según corresponda. (20)
2. Justificar: (10)
 - a. Qué elementos del modelo es necesario persistir y cuáles no (**atención con esto**)
 - b. Cómo resolvió los impedance mismatches.
 - c. Las estructuras de datos que deban ser desnormalizadas, si corresponde.