



Olimpiada de Informática - O! SANSI 2025

1. Presentación

El Comité de la Olimpiada de Informática de San Simón convoca a estudiantes de nivel primaria y secundaria del Sistema de Educación Regular del Estado Plurinacional de Bolivia a participar de la Olimpiada de Informática San Simón 2025 (SanSInf).

La SanSInf es una competencia para estudiantes de primaria y secundaria con interés en la Informática.

2. Objetivos

Los objetivos de la Olimpiada de Informática de San Simón son:

- Motivar el interés por la Informática como ciencia e ingeniería en los estudiantes de primaria y secundaria.
- Identificar y preparar a los participantes con capacidades en informática competitiva para el desarrollo de sus destrezas y conocimientos.
- Fomentar el desarrollo científico y tecnológico en Bolivia.
- Promover valores humanos de respeto, solidaridad, responsabilidad, equidad y amistad entre los participantes.
- Promover el intercambio de conocimientos y experiencias de programación entre los participantes.

3. Requisitos

Ver **Convocatoria general O!SanSi**.

4. Categorías

- **Guacamayo:** Estudiantes de quinto a sexto de primaria con conocimientos en Pensamiento Computacional.
 - Resolución de problemas con pensamiento computacional al estilo sin codificación (no se escribe código de programación), Informática desenchufada (no se usa lenguajes de programación ni computadoras para procesar las soluciones)



- **Bufo:** Estudiantes de primero a tercero de secundaria que tengan conocimiento de programación básica en C/C++ o Python.
 - Resolución de problemas computacionales haciendo uso de algoritmos y lenguajes de programación.
- **Puma:** Estudiantes de cuarto a sexto de secundaria que tengan conocimiento de programación estructurada básica en los lenguajes C/C++ o Python.
 - Resolución de problemas computacionales haciendo uso de algoritmos y lenguajes de programación.

5. Etapas

1. **Etapla Clasificatoria-** Las evaluaciones se realizan de forma presencial en el campus de la FCyT-UMSS.
2. **Etapla Final-** Las evaluaciones se realizan de forma presencial en el campus de la FCyT-UMSS.

Para las fechas de competencia ver **Convocatoria general O!SanSi**

6. Contenidos

Categoría Guacamayo

- **Abstracción:** Enfocarse en los datos e información útil e importante, aprendiendo a ignorar los detalles irrelevantes y de adorno.
- **Pensamiento Algorítmico:** manejo de actividades en secuencia y reglas para ejecutar paso a paso la solución de un problema. Se debe ser capaz de crear la secuencia y las reglas.
- **Descomposición:** pensar en términos de componentes. Los problemas para resolverlos se pueden dividir en tareas que luego se deben integrar para obtener una solución integral. El concepto de “dividir para vencer”
- **Reconocimiento de patrones:** Extrapolación en interpolación de datos, iteración de secuencias y reglas, reconocimiento de similitudes y conexiones.
- **Evaluación:** Reconocer la mejor solución entre varias. Pensar en término del uso de recursos y el tiempo.

Categoría Bufo

- Resolución de problemas: lógicos, aritmética, álgebra.
- Definición de datos: Tipos de datos., Constantes y Variables.
- Construcción de algoritmos: Expresiones y asignaciones, Entrada y salida, Estructuras de control condicionales, Estructuras de control iterativas.



- Codificación de algoritmos: Sintaxis básica y semántica, Entrada y salida de datos, Estructuras de control (Condicionales: if-then-else, switch, Iteraciones: for, while, do-while).
- Manejo de Series: aritméticas, geométricas, especiales.
- Operaciones con números: Descomposición de números, Máximo Común Divisor (MCD), Mínimo Común Múltiplo (mcm), factorial.
- Números Primos: identificar un número primo.
- Divisibilidad
- Aritmética modular
- Estrategia “descomposición funcional” (uso de funciones).
- Construcción de Funciones y paso de parámetros.
- Estructuras de datos estáticas:
 - Cadenas: funciones básicas con cadenas, concatenación, subcadenas.
 - Vectores (Arreglos): recorridos, impresión.
 - Matrices (Arreglos bidimensionales): recorridos, impresión.
- Búsquedas: búsqueda de caracteres en cadenas, búsqueda de subcadenas en cadenas.
- Métodos de ordenación simples: por inserción, por selección.
- MATEMÁTICA:
 - Enteros, operaciones (incluida exponenciación), comparación.
 - Propiedades básicas de enteros (signo, paridad, divisibilidad).
 - Propiedades modulares básicas: adición, sustracción, multiplicación.
 - Números primos
 - Fracciones, porcentajes
 - Línea, segmento, ángulo, triángulo, rectángulo, cuadrado, círculo.
 - Punto, vector, coordenadas en el plano.
 - Polígonos (Vértices, lado/arista, concavidad, convexidad, área).
 - Distancia euclidiana.
 - Teorema de Pitágoras.
 - Sistemas de numeración (binarios y n-arios).
- ESTRUCTURAS DISCRETAS:
 - Conjuntos (Inclusión/exclusión, complementos, producto cartesiano, subconjuntos).
 - Lógica de primer orden.
 - Conectividad lógica (incluidas sus propiedades básicas).
 - Tablas de verdad.
 - Cuantificadores universales y existenciales.
 - Modus ponens y Modus tollens.
 - Conteo (regla de suma y producto, progresiones aritmética y geométrica, números Fibonacci).
 - Función factorial, coeficientes binomiales.



- Triángulo de Pascal, teorema del binomio.
- FUNDAMENTOS DE PROGRAMACIÓN:
 - Sintaxis y semántica básica de lenguajes de alto nivel
 - Tipos de variables, expresiones y asignación.
 - Entrada y salida de datos.
 - Estructuras de control iterativas y condicionales.
 - Funciones y paso de parámetros.
 - Descomposición estructurada.
 - Estrategias para resolución de problemas (planeación, separación de ideas, generalización, especialización, distinción de casos y complejidad, etc.).
 - El rol de los algoritmos en el proceso de resolución de problemas.
 - Implementación de estrategias para algoritmos.
 - Estrategias de depuración.
 - Tipos de datos primitivos (booleano, entero, carácter).
 - Arreglos (Incluyendo arreglos multidimensionales).
 - Cadenas, su procesamiento y propiedades.
 - Uso básico de números reales en problemas.
 - El punto flotante para la representación de números y existencia de problemas de precisión con el punto flotante.
 - Estrategias de dividir y vencer.
- ALGORITMOS:
 - Notación de la O.
 - Clases estándar de complejidad (Constante, logarítmica, lineal, cuadrática, etc.).
 - Estrategias simples de iteración.
 - Algoritmos de fuerza bruta.
 - Algoritmos golosos.
 - Divide y vencerás.
 - Manipulación de cadenas
 - Algoritmos simples con enteros (Criba de Eratóstenes, Algoritmo de Euclides, prueba de primalidad eficiente, factorización, exponenciación).
 - Manipulación básica de arreglos.
 - Algoritmos básicos de cadenas (búsqueda simple por subcadena).
 - Búsqueda secuencial y binaria.
 - Algoritmos de ordenamiento.
- ESTRUCTURAS DE DATOS:
 - Colas, pilas, mapas, listas y conjuntos.

Categorías Puma

- Resolución de problemas: lógicos, aritmética, álgebra.



- Definición de datos: Tipos de datos., Constantes y Variables.
- Construcción de algoritmos: Expresiones y asignaciones, Entrada y salida, Estructuras de control condicionales, Estructuras de control iterativas.
- Codificación de algoritmos: Sintaxis básica y semántica, Entrada y salida de datos, Estructuras de control (Condicionales: if-then-else, switch, Iteraciones: for, while, do-while).
- Manejo de Series: aritméticas, geométricas, especiales.
- Operaciones con números: Descomposición de números, Máximo Común Divisor (MCD), Mínimo Común Múltiplo (mcm), factorial.
- Números Primos: identificar un número primo.
- Estrategia “descomposición funcional” (uso de funciones).
- Construcción de Funciones y paso de parámetros.
- Estructuras de datos estáticas:
 - Cadenas: funciones básicas con cadenas, concatenación, subcadenas.
 - Vectores (Arreglos): recorridos, impresión.
 - Matrices (Arreglos bidimensionales): recorridos, impresión.
- Búsquedas: búsqueda de caracteres en cadenas, búsqueda de subcadenas en cadenas.
- Métodos de ordenación simples: por inserción, por selección.
- Construcción de Tipos de Datos Abstractos. Por ejemplo, estructuras estáticas (struct en C++), objetos.
- Estructuras de datos dinámicas lineales: Vector, Pila, Cola, Manejo de estructuras con bibliotecas.
- Operaciones de bits: not, and, or, xor, left shift, right shift.
- Teoría de números: factores primos, criba de Eratóstenes.
- Búsqueda de patrones en cadenas.
- Combinatoria: permutaciones, combinaciones, coeficiente binomial.
- Teoría de grafos: Representación de un grafo con listas de adyacencia, Recorrido por amplitud (BFS), Recorrido por profundidad (DFS).
- MATEMÁTICA:
 - Enteros, operaciones (incluida exponenciación), comparación.
 - Propiedades básicas de enteros (signo, paridad, divisibilidad).
 - Propiedades modulares básicas: adición, sustracción, multiplicación.
 - Números primos
 - Fracciones, porcentajes
 - Línea, segmento, ángulo, triángulo, rectángulo, cuadrado, círculo.
 - Punto, vector, coordenadas en el plano.
 - Polígonos (Vértices, lado/arista, concavidad, convexidad, área).
 - Distancia euclidiana.
 - Teorema de Pitágoras.



- Sistemas de numeración (binarios y n-arios).
- No se incluirán problemas de geometría computacional (Dualidad punto línea, Hulls, Voronoi, Sweep Line, etc).
- ESTRUCTURAS DISCRETAS:
 - Funciones (inyectiva, sobreyectiva, biyectiva, composición, inversas).
 - Relaciones (Reflexividad, simetría, transitividad, equivalencia, relaciones totales o lineares, orden lexicográfico).
 - Conjuntos (Inclusión/exclusión, complementos, producto cartesiano, subconjuntos).
 - Lógica de primer orden.
 - Conectividad lógica (incluidas sus propiedades básicas).
 - Tablas de verdad.
 - Cuantificadores universales y existenciales.
 - Modus ponens y Modus tollens.
 - Técnicas de pruebas.
 - Nociones de implicación, inverso, contra-positivo, negación y contradicción.
 - Prueba directa por: contra-ejemplo, contraposición y contradicción.
 - Inducción matemática.
 - Definiciones recursivas matemáticas.
 - Conteo (regla de suma y producto, progresiones aritmética y geométrica, números Fibonacci).
 - Permutaciones y combinaciones.
 - Función factorial, coeficientes binomiales.
 - Principio de Inclusión-exclusión.
 - Principio de palomar.
 - Triángulo de Pascal, teorema del binomio.
 - Árboles y sus propiedades básicas.
 - Grafos no dirigidos (Grado, caminos, ciclos, conectividad, camino/ciclo Euleriano-Hamiltoniano, lema del apretón de manos).
 - Árboles de expansión.
 - Estrategias de recorrido.
 - Grafos decorados con distancias en aristas, pesos y colores.
 - Multígrafos con auto-ejes.
 - Grafos bipartitos.
 - Grafos planares.
- FUNDAMENTOS DE PROGRAMACIÓN:
 - Sintaxis y semántica básica de lenguajes de alto nivel
 - Tipos de variables, expresiones y asignación.
 - Entrada y salida de datos.



- Estructuras de control iterativas y condicionales.
- Funciones y paso de parámetros.
- Descomposición estructurada.
- Estrategias para resolución de problemas (planeación, separación de ideas, generalización, especialización, distinción de casos y complejidad, etc.).
- El rol de los algoritmos en el proceso de resolución de problemas.
- Implementación de estrategias para algoritmos.
- Estrategias de depuración.
- Conceptos y propiedades de algoritmos (eficiencia, correctitud).
- Tipos de datos primitivos (booleano, entero, carácter).
- Arreglos (Incluyendo arreglos multidimensionales).
- Cadenas, su procesamiento y propiedades.
- Alocaación de memoria estática y en pila.
- Implementación de estrategias para grafos y árboles.
- Estrategias para escoger una estructura de datos apropiada.
- Uso básico de números reales en problemas.
- El punto flotante para la representación de números y existencia de problemas de precisión con el punto flotante.
- Punteros y referencias.
- Uso de fracciones para cálculos precisos.
- Recursividad.
- Concepto de recursividad.
- Funciones matemáticas recursivas.
- Procedimientos recursivos simples.
- Estrategias de dividir y vencer.
- Implementación de recursión.
- Backtracking recursivo.
- Memoización de estructuras recursivas.
- Estrategias para problemas interactivos con el juez.
- Manejo de bits, máscara de bits y distintas operaciones con bits (AND, OR, XOR, etc.)
- ALGORITMOS:
 - Notación de la O.
 - Clases estándar de complejidad (Constante, logarítmica, lineal, cuadrática, etc.).
 - Estrategias simples de iteración.
 - Algoritmos de fuerza bruta.
 - Algoritmos golosos.
 - Divide y vencerás.
 - Backtracking.



- Programación dinámica en general.
- Algoritmos simples con enteros (Criba de Eratóstenes, Algoritmo de Euclides, prueba de primalidad eficiente, factorización, exponenciación).
- Manipulación básica de arreglos.
- Algoritmos básicos de cadenas (búsqueda simple por subcadena).
- Búsqueda secuencial y binaria.
- Algoritmos de ordenamiento.
- Recorrido por anchura (BFS) y recorrido por profundidad (DFS) para recorridos de grafos.
- Aplicación de DFS para hallar orden topológico y caminos de Euler.
- Encontrar componentes conexos y puentes.
- Algoritmos para hallar el camino más corto (Dijkstra, Bellman-Ford, Floyd-Warshall).
- Árbol de expansión mínima (Prim, Kruskal).
- Algoritmos para computar el emparejamiento bipartito máximo.
- Biconectividad en grafos no dirigidos (puentes, puntos de articulación).
- Conectividad en grafos dirigidos (componentes fuertemente conexos).
- Conceptos básicos de la teoría combinatoria de juegos.
- ESTRUCTURAS DE DATOS:
 - Colas, pilas, mapas, listas y conjuntos.
 - Representación de grafos (lista de adyacencia, matriz de adyacencia).
 - Representación de conjuntos-disjuntos: Union Find.
 - Árboles binarios balanceados estáticos, incluyendo Fenwick tree y Segment tree.
 - Árboles binarios balanceados.
 - Algoritmos en $O(\log n)$ para encontrar el ancestro común más bajo de dos nodos en un árbol estático.
 - Estructuras anidadas, como ser un conjunto lleno de conjuntos.
 - Tries.

Entornos y Tecnologías

Categoría Guacamayo

- Lápiz y papel.

Categoría Bufe y Puma

- Ubuntu 22.04 LTS (ubuntu-desktop-minimal package set)
(<https://ubuntu.com/download/desktop>)
- Compilador: GCC 11.3 (C/C++)



- Intérprete: Python 3.10.6
- Editores:
 - GNU Emacs 27.1
 - Visual Studio Code 1.78.2 (with C/C++ extension v1.15.4)
 - Geany 1.38
 - Vim 8.2.4919
 - Atom 1.60.0
- Depuradores:
 - DDD 3.3.12
 - GDB 12.1
 - Valgrind 3.18.1

7. Premios y Ganadores

La calificación es sobre 100 puntos.

Se clasifican 60 competidores en cada categoría a la etapa final.

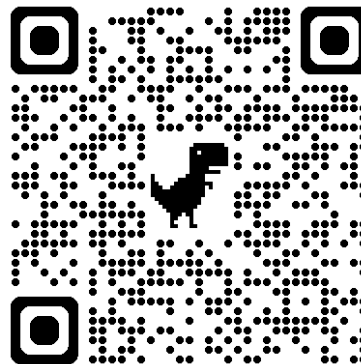
Los criterios de clasificación son:

- Los 6 mejor puntuados clasifican directamente.
- De los restantes competidores se clasifican los 6 mejor puntuados de cada departamento.
- En caso de no existir suficientes competidores en un departamento, las plazas restantes se distribuye entre los mejores puntuados que restan clasificar.
- La puntuación mínima para clasificar es de 25 sobre 100 puntos.

Se otorga mención de honor a los participantes con una puntuación superior a 25 y mayor al promedio de los participantes que no alcanzaron una medalla.

La posición en el medallero es influida por el tiempo de entrega de cada tarea o problema, Para la categoría Puma existe una penalización de 20 minutos por cada entrega con respuesta incorrecta ó que excede en el tiempo de ejecución deseado.

8. Material de estudio





Categoría Guacamayo

- https://drive.google.com/drive/folders/1I8qoJgTPFlo9x2Ii5PDof1IJHZoAle?usp=drive_li nk

Categoría Bufe

- Robert Sedgewick. Colaborador(es): Davara Rodríguez, Fernando [tr.]. Algoritmos en C++. Addison-Wesley Iberoamericana : Díaz de Santos, 1995. ISBN: 0201625741
- Manual Olimpiada Informática Argentina (<https://www.oia.unsam.edu.ar/wp-content/uploads/2022/06/Manual-OIA-2022.pdf>)
- Skiena, S. S. (2020). The algorithm design manual. Springer International Publishing.
- Sedgewick, R., & Wayne, K. (2011). Algorithms (4th edition).
- Halim, S., Halim, F., Skiena, S. S., & Revilla, M. A. (2013). Competitive programming 3.
- Laaksonen, A. (2017). Competitive Programmer's Handbook. Preprint.
- Introductory Computer Science: Bits of Theory and Bytes of Practice. A. K. Dewdney. Computer Science Press, 1996
- USACO GUIDE (<https://usaco.guide/>)

Dónde entrenar:

- <https://www.codechef.com/>
- <https://projecteuler.net/>
- <https://open.kattis.com/>
- <https://codeforces.com/>
- <https://usaco.training/>

Categoría Puma

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, y Clifford Stein. Introduction to Algorithms
- Steven Halim, Felix Halim y Suhendry Effendy. Competitive Programming 4.
- Capítulo ACM UMSA. Introducción en C++ a la programación competitiva,
- Antti Laaksonen. Competitive Programmer's Handbook.
- Skiena, S. S. (2020). The algorithm design manual. Springer International Publishing.
- Ian Parberry and William Gasarch. Ian Parberry, 2002. Problems on Algorithms (2nd Edition).
- Solutions to problems from IOI 2018 (<https://ioinformatics.org/files/booklet-ioi2018.pdf>)
- Halim, S., Halim, F., Skiena, S. S., & Revilla, M. A. (2013). Competitive programming 3.



- Andrej Vitek, Iztok Tvrdy, Robert Reinhardt, Bojan Mohar, Marc Martinec, Tomi Dolenc and Vladimir Batagelj. Problems in Programming: Experience through Practice. John Wiley & Sons, 1991.
- Steven Skiena and Miguel Revilla. Programming Challenges: The Programming Contest Training Manual. Springer-Verlag, 2003.
- USACO GUIDE (<https://usaco.guide/>)

Dónde entrenar:

- <https://www.codechef.com/>
- <https://projecteuler.net/>
- <https://open.kattis.com/>
- <https://codeforces.com/>
- <https://usaco.training/>

9. De la Organización

La Olimpiada de Informática SanSInf es organizada y dirigida por el Comité Académico/Científico del Departamento de Informática y Sistemas de la Universidad Mayor de San Simón. La Olimpiada se co-organiza con la Olimpiada Boliviana de Informática (OBI).



ANEXOS

Reglamentación sobre fraude

Las y los competidores deben utilizar únicamente una computadora y la cuenta que se les asigne en la competencia, en particular:

- Los competidores no pueden usar, enviar, copiar código que no sea de su autoría (códigos de otros competidores, código generado por Inteligencia Artificial).
- Los competidores no deben intentar enviar programas ilegales (programas que utilizan librerías para acceso a red, lectura de archivos que comprometen el sistema operativo o dan acceso a información de los problemas planteados y sus soluciones, código orientado a producir mal funcionamiento y/o compromiso del sistema operativo), ni intentar causar daño e interferencia o compromiso con el sistema de evaluación.
- Los competidores no deben intentar tener acceso a niveles superiores a los de competidor (Superusuario, comité de evaluación, etc.), o cualquier otra cuenta que no se les asignó.
- Los competidores no deben intentar almacenar información en cualquier parte del sistema de archivo, salvo en el escritorio, directorio base de su cuenta, o directorio temporal (/temp)
- Los competidores no deben tocar ni acceder a otras computadoras y sus accesorios, salvo la que se les asignó.
- Los competidores no deben intentar acceder a otras máquinas/dispositivos en la red o internet, salvo lo requerido, para enviar su solución (respuesta) al sistema de evaluación.
- Los competidores no deben intentar reiniciar o alterar la secuencia de inicio y configuración del computador que se les asigna.
- Los competidores no deben comunicarse con otras personas durante la competencia, salvo personal del equipo asignado por el Comité a cargo de la competencia y/o miembros del Comité Científico.
- Los competidores no deben ni pueden ingresar dispositivos electrónicos de almacenamiento y/o transmisión de datos a la competencia, así mismo, no pueden ingresar ningún tipo de material impreso.
- Los competidores no pueden copiar ni enviar código que no sea de su autoría (códigos de otros competidores o código generado por Inteligencia Artificial)
- Los competidores pueden ingresar papel (completamente limpio, sin escrituras), y block de notas para escribir.

El incumplimiento del presente reglamento, será considerado como fraude y resultará en la descalificación del o la estudiante infractor(a).



Código de conducta

El código de conducta de la Olimpiada Informática se basa en el código de conducta de la Olimpiada Internacional de Informática (<https://ioinformatics.org/page/code-of-conduct/50>)

La comunidad de la Olimpiada es una comunidad diversa en la que los estudiantes, tutores, y miembros de los comités tienen la oportunidad de interactuar con personas de diferentes culturas, etnias, religiones, discapacidades, identidades de género y orientaciones sexuales. La comunidad de la Olimpiada abraza la diversidad y promueve el respeto y la comprensión mutuos.

La Olimpiada de Informática tiene el compromiso de aceptar a todos los participantes tal como son y a brindar un entorno en el que todos los apasionados por la informática se sientan cómodos e incluidos. Para lograr este objetivo, se requiere la participación respetando este código de conducta en todas las actividades y animando a los demás a hacer lo mismo.

Los participantes deben ser considerados tolerando la diversidad cultural, mantener un diálogo respetuoso cuidando las acciones y palabras. Si ves que alguien se siente incómodo con lo que dices o haces, detente y determina si lo que estás haciendo es apropiado y si se puede hacer de una manera diferente y más respetuosa.

Piensa y actúa de forma inclusiva, tratando a las personas como seres humanos iguales. Se debe observar cuidadosamente si algunas personas puedan quedar excluidas u objetivadas debido a su género, cultura, religión, creencias, discapacidad u orientación sexual.

No se debe hacer bromas, comentarios o gestos ofensivos verbales o escritos relacionados con el género, la orientación sexual, la discapacidad, la apariencia física, el tamaño del cuerpo, el origen étnico o la religión, incluso a través de medios en línea como grupos de Whatsapp y otras redes sociales.

No se tolerará la intimidación, el acoso ni ningún comportamiento abusivo, discriminatorio o despectivo en la Olimpiada. Las consecuencias de tal comportamiento, dependiendo de su gravedad, podrían ir desde una simple advertencia hasta la descalificación de la competencia.

Si observa o experimenta comportamientos indebidos reñidos con el código de conducta, debe comunicarse con un miembro del Comité de la Olimpiada. Todas las quejas presentadas ante el Comité de la Olimpiada se tomarán en cuenta y los denunciantes permanecerán confidenciales.