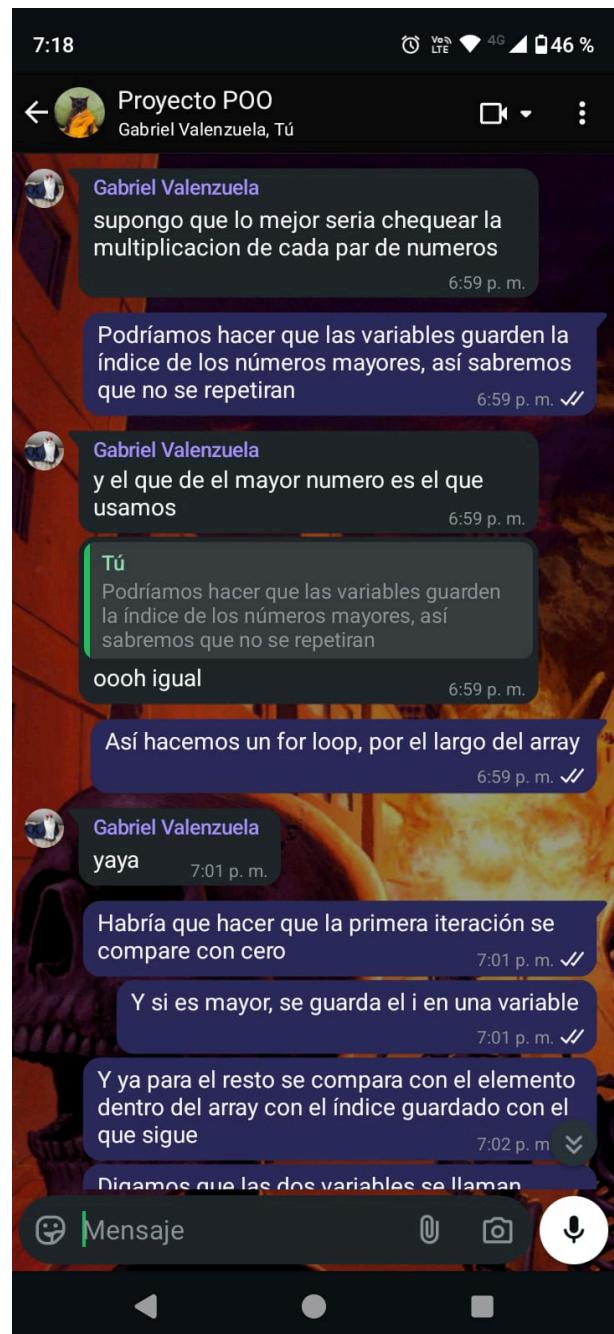
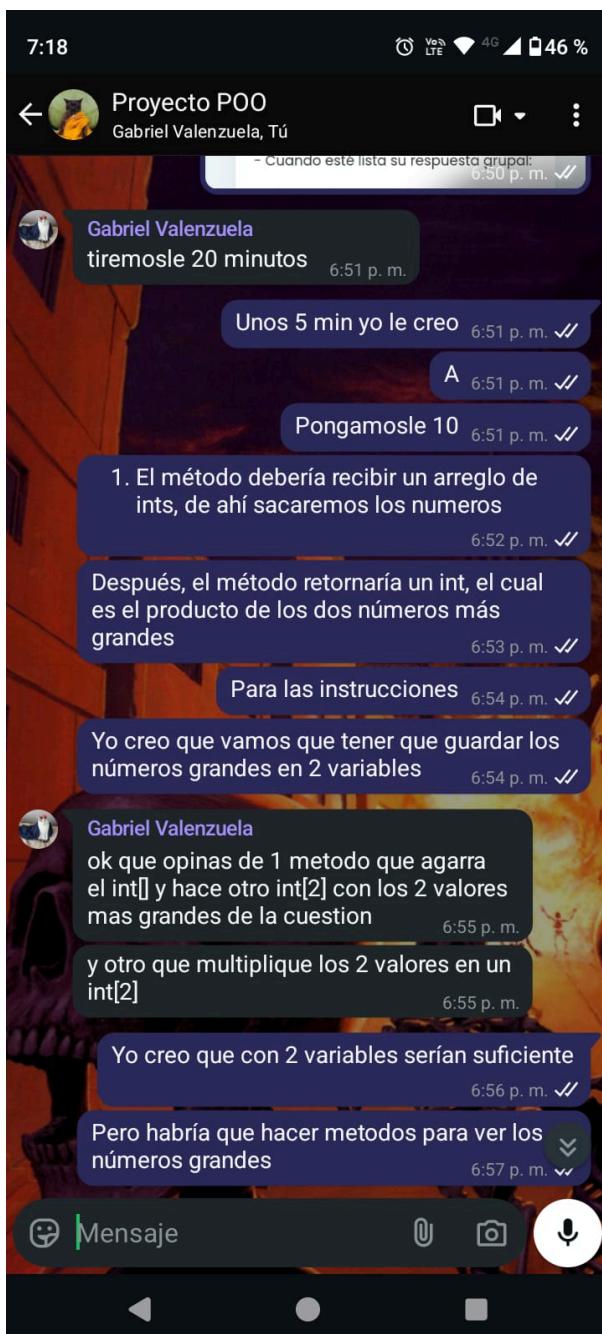


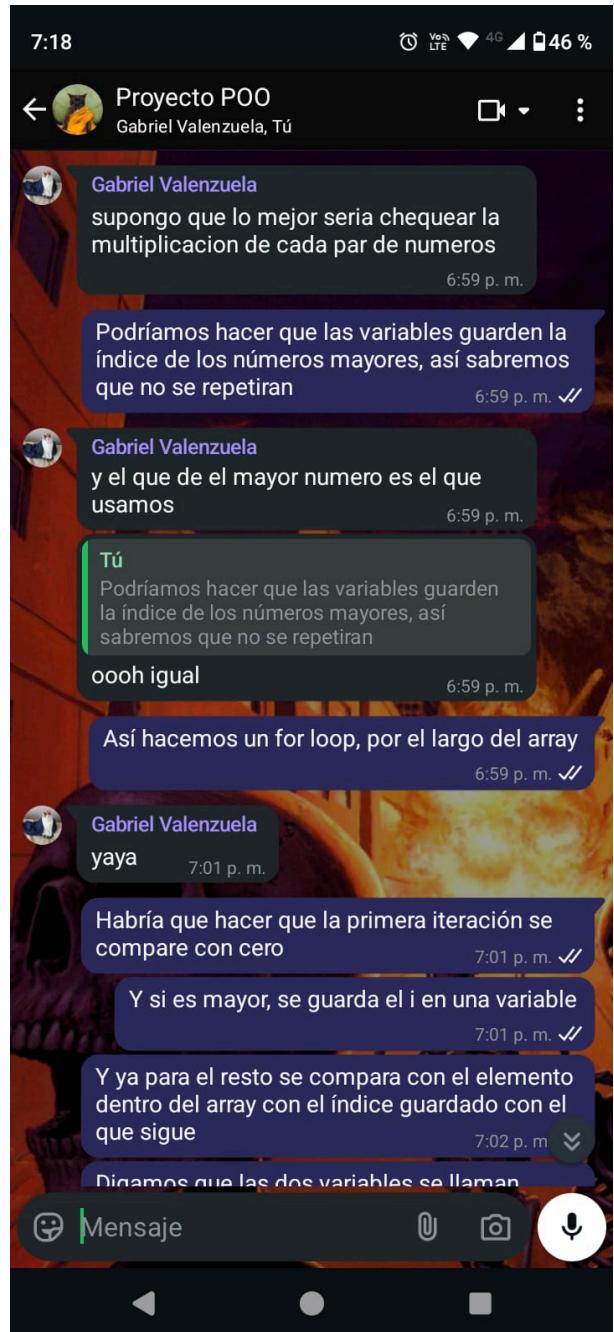
Evidencia 03: Caso Palíndromos

Por Gabriel Valenzuela y Gabriel Vergara

Actividad 0: Análisis del caso

Mi compañero Gabriel estimó que esta actividad debería demorarse unos 20 min, luego yo (Gabriel) dije unos 5 min, cambiando de parecer dije unos 10. Al final Gabriel tenía razón, nos demoramos unos 20 minutos en analizar el caso.







- Evidencia de la discusión

The screenshot shows a GitHub repository page. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A search bar is also at the top. Below the header, the repository name 'CasoDeLosNumerosAdyacentes' is shown, along with a 'Public' badge. A message from 'dev' indicating recent pushes is displayed. On the left, there's a sidebar with options for master branch, 2 branches, 0 tags, and a 'Code' button. The main content area shows a list of commits from 'MountainKing187'. One commit, 'Create README.md', is highlighted. The commits are:

- .idea Initial commit 18 minutes ago
- .gitignore Initial commit 18 minutes ago
- README.md Create README.md 17 minutes ago
- pom.xml Initial commit 18 minutes ago

Below the commits, there's a section for the 'README' file, which contains the following text:

```
Actividad en clase/evidencia de la semana - Caso de los Números adyacentes Integrantes: Gabriel Valenzuela y Gabriel Vergara

Objetivo: desarrollar una solución completa basada en métodos, que aplique las buenas prácticas al caso planteado. Además debe considerar la implementación de pruebas unitarias usando Junit y la gestión de errores a través del uso de excepciones.

Caso a resolver

Se entrega un arreglo de enteros a un método llamado productoAdyacentes. Dicho método debe retornar el mayor producto de números adyacentes que encuentre.

P. ej., dado el arreglo arreglo = {1, -4, 2, 2, 5, -1}, el mayor producto de números adyacentes es el de los números 2 y 5, en cuyo caso el método retorna el valor 10.

Luego, si se hace algo como System.out.print("El producto adyacente es: "+ productoAdyacentes(arreglo));
```

On the right side, there are sections for About (with no description), Activity (0 stars, 1 watching, 0 forks), Releases (no releases published, Create a new release), and Packages (no packages published, Publish your first package). The bottom of the page includes a footer with copyright information and a link to the repository's URL.

- Evidencia del git y del readme

Actividad 1: implemente su método

Yo (Gabriel) estimé que me demoraría en implementar el método productoAdyacentes y su prueba en 10 minutos. Y me demore unos 9 minutos.

```
public class casoAdyacentes { no usages new *
    public static int productoAdyacentes(int[] arreglo){ no usages new *
        return 0;
    }
}
```

```
public class casoAdyacentes {    no usages new*  
    public static int productoAdyacentes(int[] arreglo){    no usages new*  
        int mayorProducto = 0;  
        int productoActual;  
  
        for (int i = 0; i < arreglo.length - 1; i++) {  
            productoActual = arreglo[i] * arreglo[i+1];  
            if (productoActual > mayorProducto){  
                mayorProducto = productoActual;  
            }  
        }  
        return mayorProducto;  
    }  
}
```

```
import ...  
class casoAdyacentesTest {    new*  
    @Test    new*  
    void productoAdyacentes() {  
        int[] arreglo = {1, -4, 2, 2, 5, -1};  
        assertEquals( expected: 10, casoAdyacentes.productoAdyacentes(arreglo));  
    }  
}
```

- Evidencia implementación y prueba inicial

Actividad 2: Probando su código

Yo (Gabriel) estimé que me demoraría en implementar las pruebas del código 10 minutos y me tarde 20 minutos

hay que estimar el tiempo que nos demorariamos en diseñar e implementar unas 4 pruebas

Yo cccreo que nos demorariamos unos 20 min en la actividad, 10 en el diseño y el otro 10 en la implementacion

Dev va #2

Merged by MountainKing187 merged 2 commits into dev from devVa 1 hour ago

May 8, 2025, 8:51 PM GMT-4

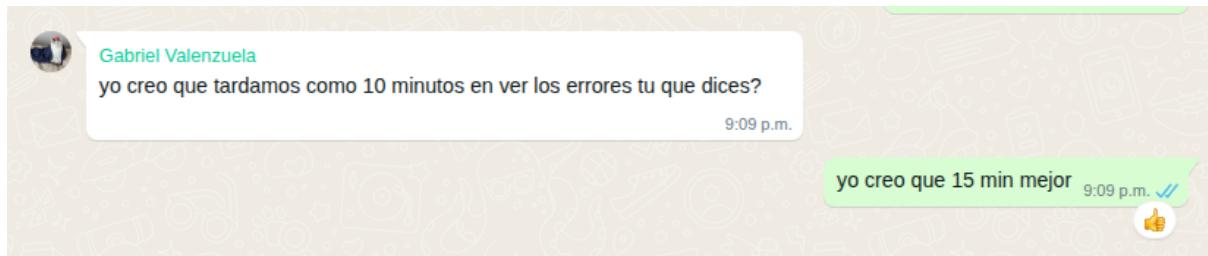
Conversation 0 Commits 2 Checks 0 Files changed 2

Tenemos 5 pruebas diseñadas.

1. arrayde2: Se le entregará un array con dos elementos para ver el funcionamiento del for loop se el correcto, el array entregado es {2,4}, lo cual debería retornar 8.
2. arraydetodoigual: Se le entregará un array con un número repetido para todos los elementos, esta prueba está diseñada para ver el correcto funcionamiento de la comparación con resultados repetidos. Para esta prueba, se le entregó {5,5,5,5} como array y debería retornar 25
3. array binario: Se le entregará un array de tamaño impar con 0 en los elementos de índice par, esta prueba está diseñada para ver si el 0 al iniciar la variable no afecta la comparación con otros cero. Para esta prueba, se le entrego {1,0,1,0,1} como array y debería entregar 0
4. arraynegativo: Se le entregará un array con todos los elementos en negativo, esta prueba está diseñada para ver si funciona la multiplicación en negativo. Para esta prueba, se le entregó {-3,-4,-6,-5,-2} como array y debería entregar 30
5. arraymixto: Se le entregará un array con elementos positivos y negativos alternados, esta prueba está diseñada para chequear que el resultado pueda ser negativo. Para esta prueba, se le entregó {-2,2,-6,5,-3} como array y debería entregar -4

Actividad 3: Garantizando el éxito

Nosotros (Gabrieles), estimamos que nos demoramos 10 minutos en identificar y solucionar los errores posibles e implementar las soluciones, nos tardamos como 30 minutos



Hemos identificado 4 errores que podrían pasar:

1. Si el array contiene una cantidad mixta de negativos y positivos, puede que el mayor número sea negativo, y en el código pusimos como la comparación a inicial con 0, por lo que siempre retorna 0 si fuera este caso. Para solucionar este error, se le pone al valor inicial como Integer.MIN_VALUE en vez de 0, por lo que tenemos entendido, siempre va a ser el valor más pequeño en una comparación.
2. Si el array es null, el programa no tenía cómo manejarlo, se le agregó una validación que chequea si el array es null y si no envia una excepción indicando el error
3. Si el array es más pequeño o más grande que los límites establecidos, estaría rompiendo con las reglas establecidas de la actividad. Por lo que se ve si el array entregado tiene al menos 2 elementos y a lo más 20, y si no es así, se enviará una excepción al método indicando el error.
4. El método multiplicaba los valores dentro del array sin importar su tamaño, para prevenir que los valores fueran extremos ahora chequea que el array no posea valores sobre 1000 o bajo -1000 y en caso de presentarse alguno enviará una excepción indicando el error
5. El programa no dejará de funcionar aunque se reconocieran que habían errores, para solucionar se agregó un try catch al método principal para que detenga el programa si encuentra un error al validar el array

Link repositorio:

<https://github.com/MountainKing187/CasoDeLosNumerosAdyacentes>