

SISTEMA DE OPERACIONES CRUD APLICADO A UN SISTEMA DE ASIGNACION DE CLASES OPTATIVAS

CRUD Operations System Applied to an Optional Assignment System

Sistema de Operaciones CRUD Aplicado a un Sistema de Asignación de Clases Optativas

**Samuel Armando Arce Suárez**

Sarce@poligran.edu.co

Politécnico Grancolombiano

**Juan David Torres Caicedo**

jdavidtorres2@poligran.edu.co

Politécnico Grancolombiano

No. ORCID (solo número)

Colombia

(incluir breve bibliografía del autor)

*Recepción:*  *XX.XX.XXXX*

*Aceptación:*  *XX.XX.XXXX*

**DOI:** *XXXXXX*

# Resumen

Este apartado debe estar diseñado para invitar al lector a continuar leyendo el documento de tal manera que se debe indicar: la motivación y propósito del trabajo, el método para llevar a cabo lo que va del documento y una idea de los resultados logrados. La extensión máxima es de 300 palabras, y puede dividirse de acuerdo con los siguientes apartados: Objetivo, Métodos y materiales, Resultados y Conclusiones (generalidades)

# Palabras clave

En este apartado se deben enunciar de **3 a 6 palabras o descriptores** asociados al contenido del manuscrito. Idealmente en este apartado se deben incluir términos universalmente categorizados y organizados en el área de conocimiento en el cual se publica; estos términos o tesauros -como son conocidos- facilitan la indización del artículo en bases de datos y buscadores especializados por ello se recomienda que no sea una transcripción separada por comas del título del documento. Todas **deben ser escritas en minúscula y separadas por coma** (,).

# INTRODUCCIÓN

Una posible operación simple, realista y poco común en una universidad que requiera un CRUD para construir una aplicación en JAVAFX es la gestión de las asignaturas optativas de los estudiantes. Esta operación permitiría a los estudiantes crear, leer, actualizar y eliminar sus preferencias de asignaturas optativas para el próximo semestre, así como consultar la disponibilidad de plazas y el horario de cada asignatura.

Descripción:

Desarrolla una aplicación en JAVAFX que permita a los estudiantes de la universidad Ernesto Fernández gestionar sus asignaturas optativas para el próximo semestre. La aplicación debe tener las siguientes características:

La aplicación debe conectarse a una base de datos que almacene la información de las asignaturas optativas, los estudiantes y sus preferencias.

La aplicación debe permitir a los estudiantes autenticarse con su nombre de usuario y contraseña.

La aplicación debe mostrar una lista de las asignaturas optativas disponibles, con su nombre, código, créditos, profesor, horario y número de plazas libres.

La aplicación debe permitir a los estudiantes seleccionar hasta tres asignaturas optativas de su interés, ordenadas por prioridad.

La aplicación debe validar que las asignaturas optativas seleccionadas no tengan conflictos de horario ni superen el límite de créditos establecido por la universidad.

La aplicación debe permitir a los estudiantes crear, leer, actualizar y eliminar sus preferencias de asignaturas optativas, así como consultar el estado de su solicitud.

La aplicación debe asignar las plazas de las asignaturas optativas según el orden de prioridad y la fecha de solicitud de los estudiantes, respetando el límite de plazas de cada asignatura.

La aplicación debe notificar a los estudiantes el resultado de su solicitud, indicando las asignaturas optativas que les han sido asignadas y las que no.

La aplicación debe tener una interfaz gráfica de usuario intuitiva, atractiva y adaptada al tamaño de la pantalla.

# DEFINICIONES

**GitHub:** GitHub es un servidor de alojamiento o repositorio en línea que permite a usuarios almacenar y colaborar en proyectos enlazados y basados en Git, un sistema para control de versiones que permite tener un registro del progreso y cambios en un proyecto con el tiempo.

**StarUML**: StarUML es un software de modelado ágil y conciso usando diagramas de flujo y Lenguaje Unificado de Modelamiento (UML) el cual es un lenguaje de modelado visual que es usado para modelar sistemas orientados a objetos, crear un lenguaje legible por humanos y máquinas y orientar al desarrollo de escala en sistemas difíciles.

**Visual Studio Code:** Editor de código ligero incluyendo compatibilidad integrada a JavaScript, TypeScript y Node.js y por medio de extensiones, es capaz de crear ambientes de desarrollo para varios lenguajes de programación y entornos de ejecución. (C++, C#, Java, Python, etc...)

**Modelo-Vista-Controlador:** Es un patrón que busca reducir el esfuerzo de programación a partir de crear un estándar en el diseño de aplicaciones. Es un paradigma que divide las partes que hacen parte de la aplicación en: Modelo, vistas y controladores. Para que sea mucho más sencilla la actualización y mantenimiento del software. El modelo es el que representa, maneja y controla los datos del programa. La vista es el que maneja la presentación visual de los datos proporcionados por el modelo. Y el controlador es el que les da un significado a las órdenes del usuario, modificando los datos que están dentro del Modelo.

**Java Collection Framework:** Es un framework que contiene varias colecciones, las cuales se pueden definir como un objeto que representa un grupo de varios objetos. Y un collection framework es una arquitectura unificada para manipular y representar varias colecciones individualmente. Consisten en: Implementaciones de propósito general, implementaciones heredadas, implementaciones para propósitos especiales, implementaciones concurrentes, implementaciones de contenedores, implementaciones de conveniencia, implementaciones abstractas, algoritmos, infraestructura, utilidades de matriz.

**Javadoc:** en Java se utiliza para generar documentación en formato HTML a partir de archivos fuente de Java. Esta documentación describe clases, interfaces, métodos, campos, constructores, etc. Puedes usarlo en archivos individuales o paquetes enteros, y puedes especificar opciones como la inclusión de clases anidadas o recursión en subpaquetes.

**JUnit:** JUnit 5 representa la última versión del marco de pruebas JUnit, proporcionando una base contemporánea para la realización de pruebas por parte de los desarrolladores en la JVM. Se centra en las versiones de Java 8 y superiores, y permite una variedad de estilos de pruebas diferentes.

# MÉTODO

Metodología:

1. - Investigar y recopilar referencias:

* Escribir la introducción, el método y citar las referencias académicas u oficiales.
* Finalizar antes del 11/03/2024.

1. Diseño de la Página en StarUML:

* Utilizar StarUML para diseñar la estructura de la página.
* Definir la arquitectura y disposición de los elementos.
* Completar antes del 23/03/2024.

1. Comienzo del Código:

* Configurar el entorno de desarrollo.
* Iniciar el desarrollo del código de la aplicación.
* Avanzar antes del 30/03/2024.

1. Desarrollo del Método y Resultados Parciales:

* Continuar desarrollando el método y documentar resultados parciales.
* Asegurarse de tener progreso significativo antes del 12/04/2024.

1. Desarrollo de la Aplicación:

* Implementar funcionalidades y realizar pruebas exhaustivas.
* Finalizar el desarrollo antes del 22/04/2024.

1. Entrega de Resultados Completos, Conclusiones y Resúmenes:

* Finalizar análisis y documentación.
* Redactar conclusiones y resumen.
* Entregar antes del 02/05/2024.

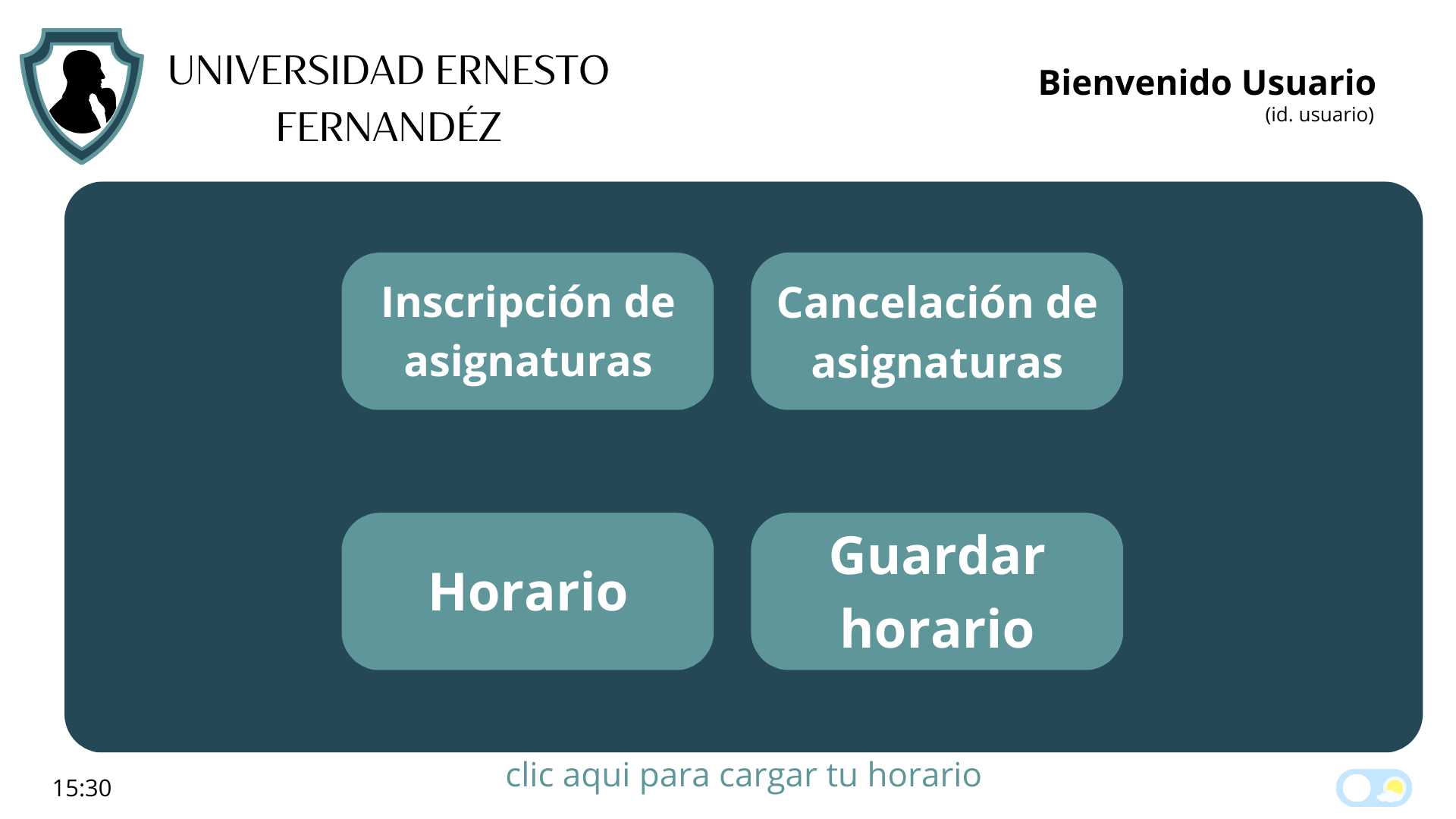
1. Revisión y Ajustes Finales:

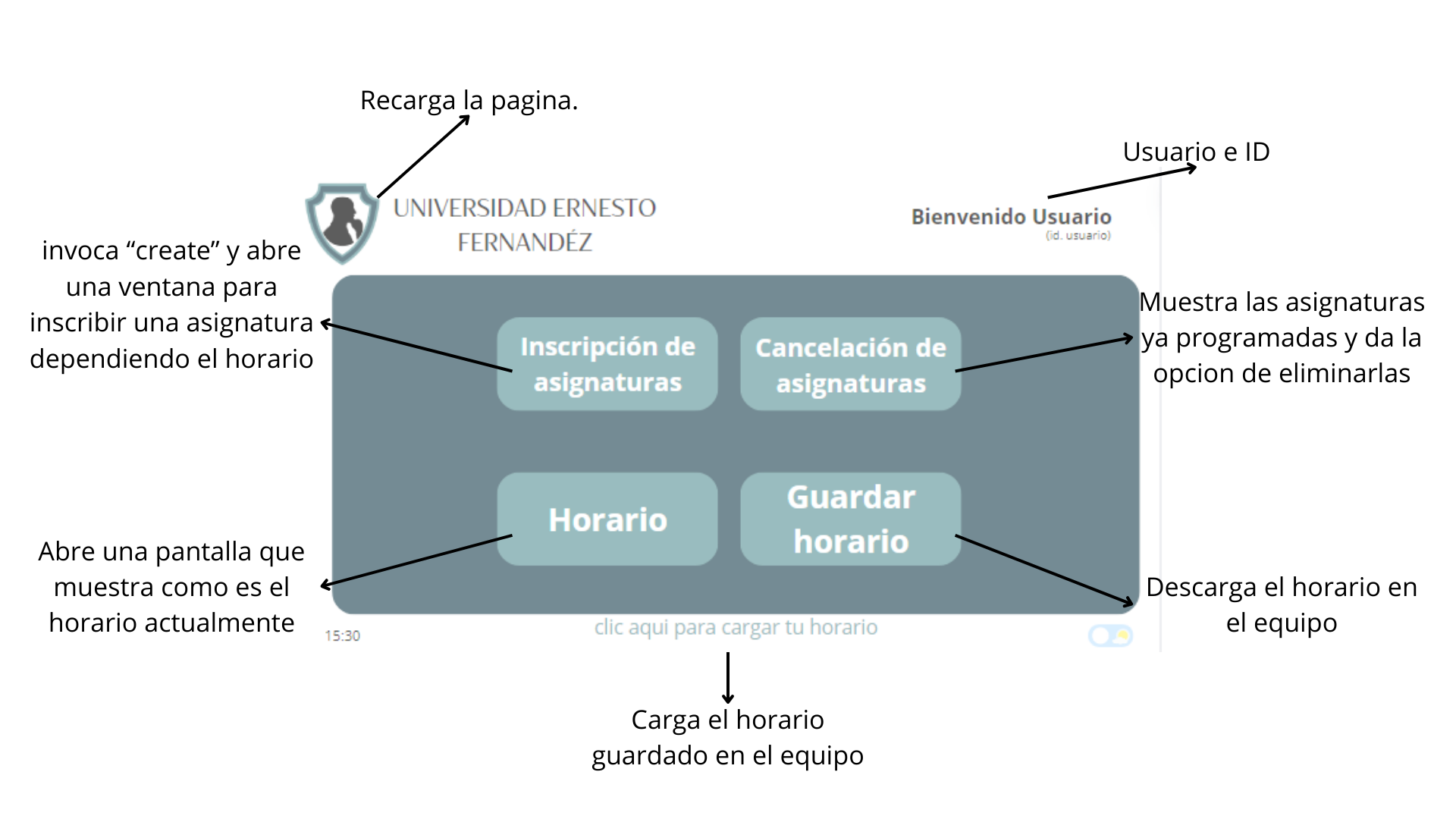
* Revisar y ajustar según sea necesario.
* Asegurarse de que todo esté completo y bien presentado.

|  |  |  |
| --- | --- | --- |
| Inicio | Descripción | Fecha de finalización |
| 16/02/2024 | Entrega introducción, método y referencias. (1,2,3) | 11/03/2024 |
| 20/02/2024 | Empezar con el diseño de la página en StarUml | 23/03/2024 |
| 27/02/2024 | Comienzo de código | 30/03/2024 |
| 12/03/2024 | Desarrollo del método (4,5), resultados parciales y desarrollo de la aplicación (6) | 22/04/2024 |
| xx/xx/xxxx | En desarrollo | xx/xx/xxxx |
| 23/04/2024 | Entrega de resultados completos, conclusiones y resúmenes (6,7). | 02/05/2024 |

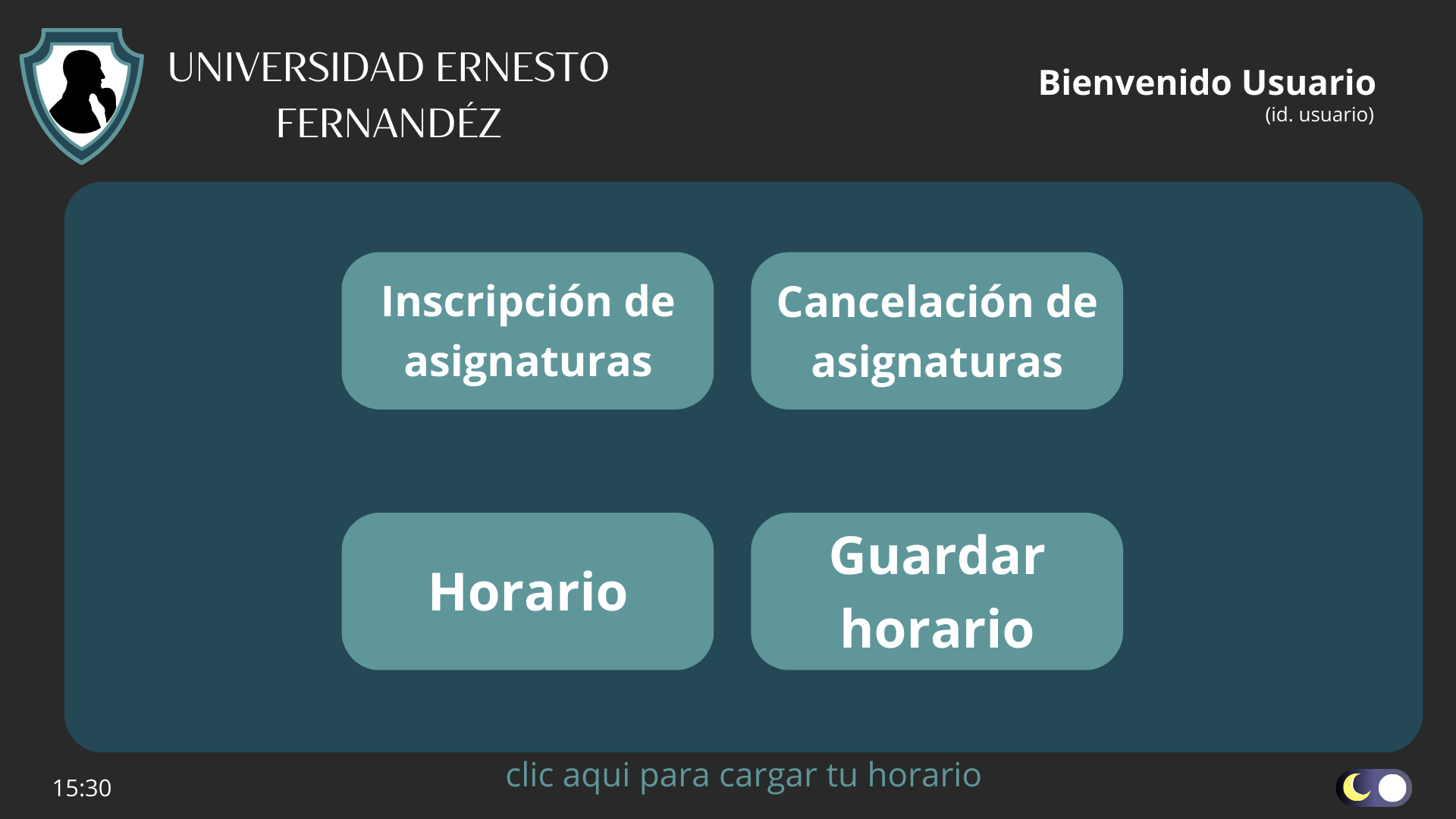
PROTOTIPO GRÁFICO DE LA APLICACIÓN:

MODO DÍA:

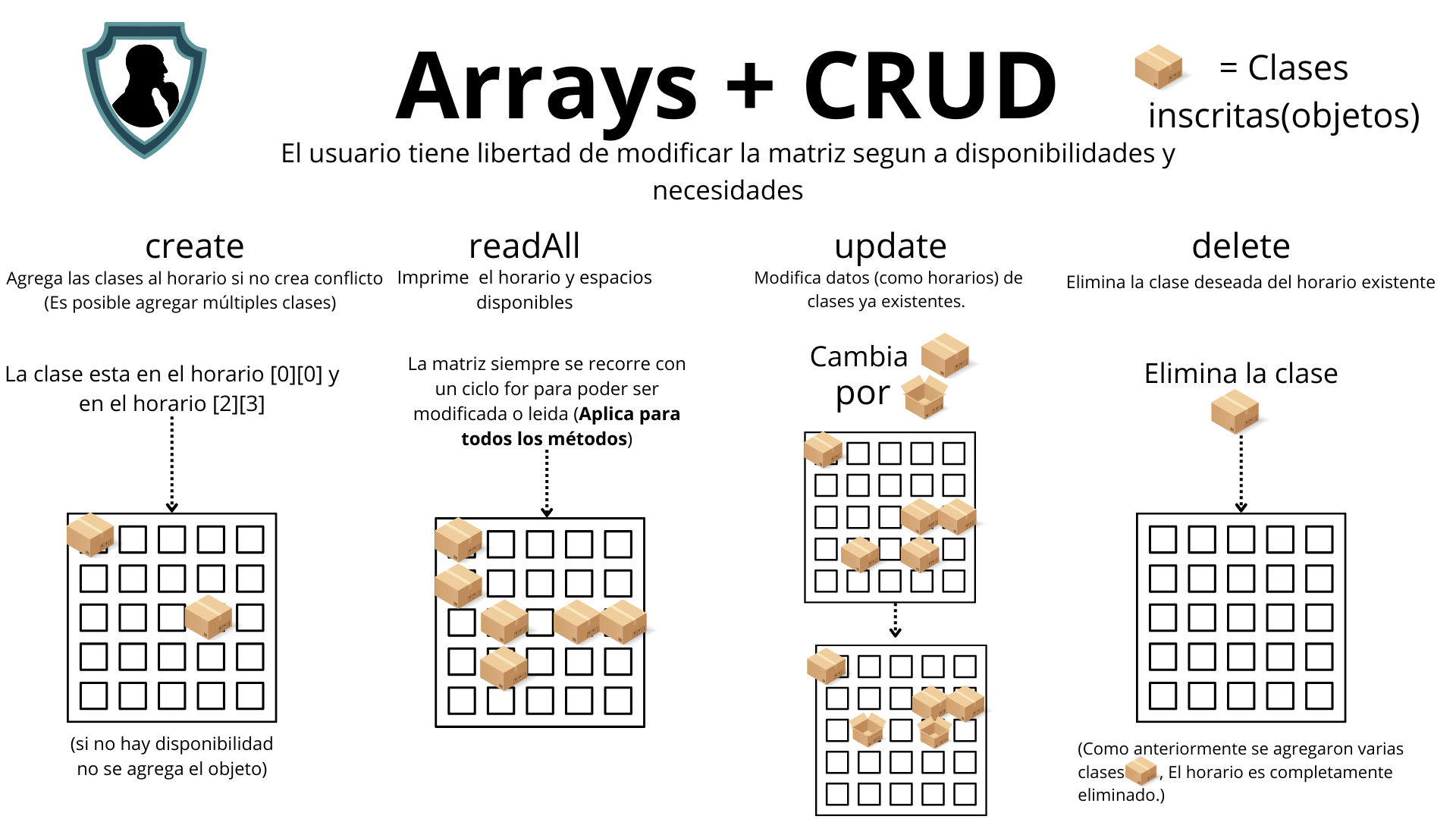
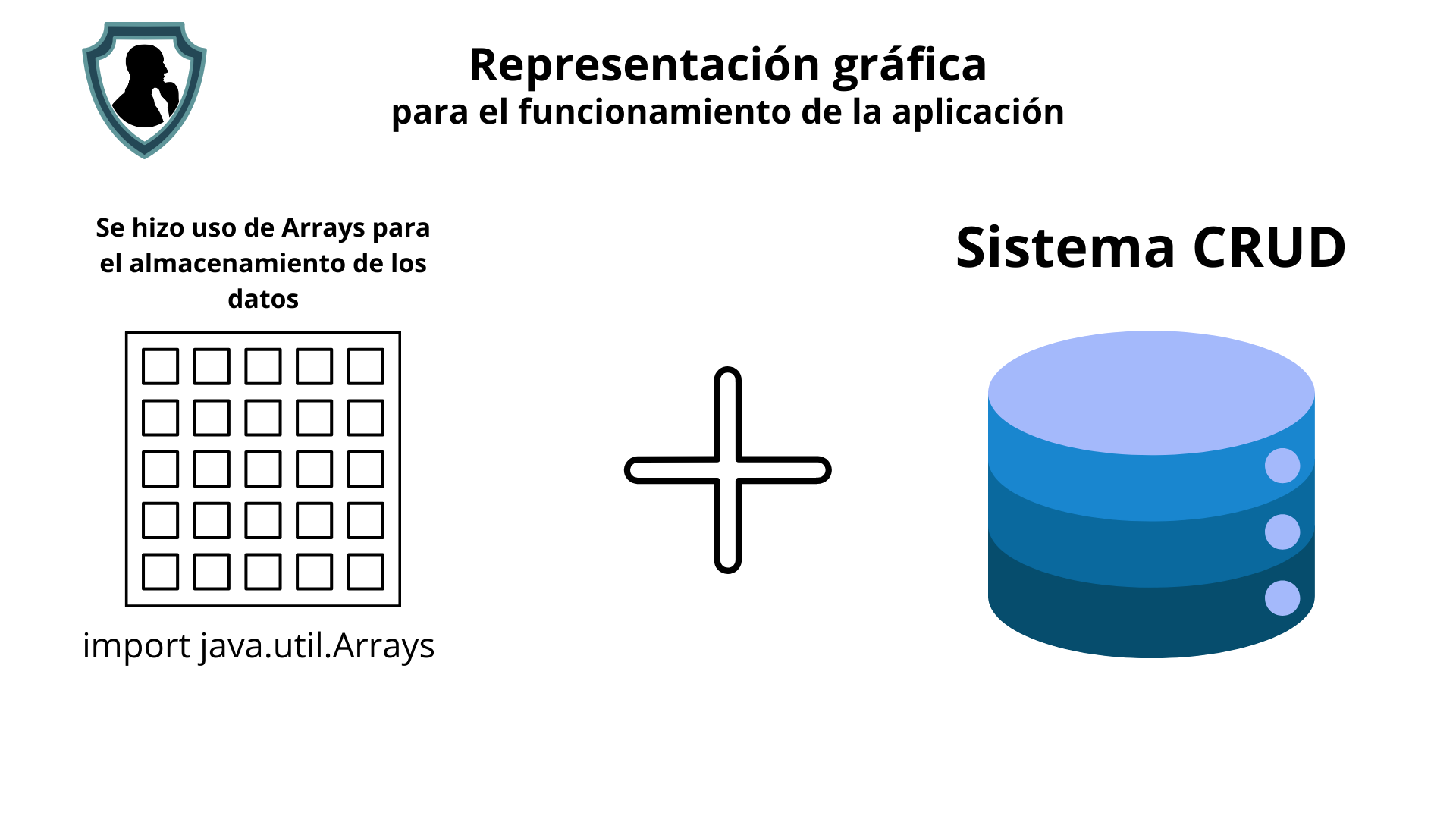




MODO NOCHE:



FUNCIONAMIENTO DE LA APLICACIÓN:



# RESULTADOS

Gracias al desarrollo, investigación y planeación del proyecto, logramos realizar una aplicación ejecutable en un IDE. Que cumple con las funciones estipuladas anteriormente (inscribir, actualizar, entre otros) y puede ser encontrada en el siguiente repositorio de GitHub: [Repositorio Proyecto Samuel Arce y Juan Torres](https://github.com/SysZ2211/ProyectoFinalPRD)

# DISCUSIÓN Y CONCLUSIÓN

Interpreta los resultados de la investigación demostrando la relevancia de la investigación e innovación en el área de conocimiento.

Necesariamente, establece recomendaciones a nuevos estudios para su abordaje o continuidad de conformidad a los resultados encontrados.

# REFERENCIAS BIBLIOGRÁFICAS

JUnit Team. (20XX). About. JUnit 5. URL: <https://junit.org/junit5/>

Anonimo,A. (20XX).Javadoc. Oracle. <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javac.html>

Anonimo, A. (20XX). Collections Framework Overview. Oracle. <https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html>

Fernández Romero, Y-Díaz González, Y. (2012, Enero-Abril). Patrón Modelo-Vista-Controlador. Telem@tica, 11, pp. 47-56)

Anonimo, A. (20XX). Overview. Visual Studio Code. <https://code.visualstudio.com/docs>Lema Peñafiel, A. (2023, Mayo-Septiembre). Análisis de la herramienta STAR UML que utiliza la metodología del lenguaje unificado de modelamiento para el desarrollo de SI.. [Tesis doctoral o de maestría, UNIVERSIDAD TÉCNICA DE BABAHOYO]. <http://dspace.utb.edu.ec/bitstream/handle/49000/14997/E-UTB-FAFI-SIST.INF-000179.pdf?sequence=1&isAllowed=y>.

Astigarraga, J- Cruz-Alonso, V. (2022, 24 Febrero). ¡Se puede entender cómo funcionan Git y GitHub!. Asociación Española de Ecología Terrestre, 31, pp. 1-4 <https://www.revistaecosistemas.net/index.php/ecosistemas/article/view/2332/1505>