

Roadmap del Proyecto

Entorno Virtual para la Enseñanza de la Biofísica

Última actualización: 30 de noviembre de 2025

Estado Actual — Versión 1.0.0

Resumen de Funcionalidades Implementadas

Módulo	Estado	Descripción
 Inicio	<input checked="" type="checkbox"/> Completo	Pantalla de bienvenida con logo y accesos rápidos
 Conferencias	<input checked="" type="checkbox"/> Completo	7 temas con 20 PDFs organizados por carpetas
 Bibliografía	<input checked="" type="checkbox"/> Completo	10 libros + 1 artículo con acceso a PDFs
 Problemas	<input checked="" type="checkbox"/> Completo	Banco de ejercicios con filtros y detalles
 Seminarios	<input checked="" type="checkbox"/> Completo	4 seminarios del curso en PDF
 Ósmosis	<input checked="" type="checkbox"/> Completo	Comparación de osmolaridades con gráficos
 Equilibrio Iónico	<input checked="" type="checkbox"/> Completo	Ecuación de Nernst con 7 iones predefinidos
 Patch Clamp	<input checked="" type="checkbox"/> Completo	Curvas I-V y registro de canal único

Características de la Interfaz

- Barra lateral con navegación visual
- Logo personalizado en sidebar, home y barra de tareas
- Paneles redimensionables con PanedWindow
- Barras de desplazamiento en resultados y gráficos
- Pestañas para organizar contenido
- Efectos hover en tarjetas de bibliografía
- Apertura de PDFs con visor del sistema

Solvers Implementados

Módulo de Ósmosis

Solver	Archivo	Funcionalidad
OsmolarityComparisonSolver	<code>osmolarity_comparison.py</code>	Comparación intra/extracelular con múltiples solutos
OsmolaritySolver	<code>osmolarity.py</code>	Cálculo de osmolaridad simple
TonicitySolver	<code>tonicity.py</code>	Clasificación de tonicidad
CellVolumeSolver	<code>cell_volume.py</code>	Volumen celular Boyle-van't Hoff
VolumeDynamicsSolver	<code>volume_dynamics.py</code>	Dinámica temporal del volumen

Módulo de Equilibrio Iónico

Solver	Archivo	Funcionalidad
NernstSolver	ionic_equilibrium/nernst.py	Potencial de equilibrio de Nernst

Módulo de Patch Clamp

Solver	Archivo	Funcionalidad
NernstSolver	patch_clamp/nernst.py	Potencial de Nernst (duplicado)
GoldmanSolver	goldman.py	Ecuación de Goldman-Hodgkin-Katz
IVCurveSolver	iv_curve.py	Curvas corriente-voltaje
SingleChannelSolver	single_channel.py	Análisis de canal único

● Prioridad Alta — Próximas Mejoras

1. Unificación de Solvers de Nernst

Estado: Pendiente

Problema: Existe duplicación entre `ionic_equilibrium/nernst.py` y `patch_clamp/nernst.py`

Tareas:

- Consolidar en un único solver base
- Mantener referencias desde ambos módulos
- Actualizar imports en las vistas

2. Integración Completa de Goldman-Hodgkin-Katz

Estado: Parcialmente implementado (solver existe, GUI incompleta)

Tareas:

- Añadir tab de Goldman en Equilibrio Iónico
- Campos para permeabilidades relativas (P_K , P_{Na} , P_{Cl})
- Gráfico comparativo con potenciales individuales de Nernst

3. Validación de Datos de Entrada

Estado: Básica

Tareas:

- Validar rangos fisiológicos de concentraciones
- Prevenir valores negativos o cero donde no aplique
- Mensajes de error específicos en español
- Validar temperatura en rangos razonables

🟡 Prioridad Media — Mejoras Funcionales

4. Exportación de Resultados

Estado: No implementado

Funcionalidades propuestas:

- Exportar resultados a PDF con formato académico
- Exportar gráficos como imágenes (PNG/SVG)
- Copiar resultados al portapapeles
- Generar reportes con interpretación completa

Archivos a crear: `src/infrastructure/export_manager.py`

5. Mejoras en Gráficos

Ubicación: `src/desktop/components/plot_canvas.py`

Mejoras propuestas:

- Zoom interactivo con scroll del mouse
- Tooltips al pasar cursor sobre puntos de datos
- Botón para guardar gráfico individual
- Personalización de colores y estilos
- Animación de cambio de volumen celular

6. Más Solutos Predefinidos

Ubicación: `src/core/solvers/osmosis/osmolarity_comparison.py`

Solutos a añadir:

- Lactato de Ringer (composición completa)
- Solución salina hipertónica (3%, 7.5%)
- Albúmina (para presión oncótica)
- Dextrosa en diferentes concentraciones

7. Calculadora de Presión Osmótica (π)

Estado: No implementado

Funcionalidad propuesta:

- Ecuación de Van't Hoff: $\pi = iMRT$
- Comparación con presión oncótica del plasma (~25 mmHg)
- Predicción de dirección del flujo de agua

Archivo a crear: `src/core/solvers/osmosis/osmotic_pressure.py`



Prioridad Baja — Mejoras de UX/UI

8. Modo Claro/Oscuro Dinámico

Estado: Solo modo oscuro

Tareas:

- Añadir toggle de tema en sidebar o configuración
- Guardar preferencia de tema
- Asegurar legibilidad en ambos modos
- Adaptar gráficos al tema seleccionado

9. Atajos de Teclado

Estado: No implementado

Atajos propuestos:

Atajo	Acción
<code>Ctrl+1-6</code>	Cambiar entre módulos principales
<code>Ctrl+Enter</code>	Ejecutar cálculo
<code>Ctrl+R</code>	Resetear formulario
<code>Ctrl+E</code>	Exportar resultados
<code>F1</code>	Mostrar ayuda contextual

10. Tooltips y Ayuda Contextual

Estado: Mínimo

Tareas:

- Tooltips explicativos en todos los campos de entrada
- Panel de ayuda con fórmulas y ecuaciones
- Enlace a recursos externos relevantes
- Explicación de unidades y rangos típicos

11. Internacionalización (i18n)

Estado: Solo español

Tareas:

- Extraer strings a archivos de traducción
- Implementar soporte para inglés
- Selector de idioma en configuración
- Traducir contenido de ayuda

Contenido a Expandir

12. Banco de Problemas

Ubicación: `data/problems/`

Categorías pendientes:

- Problemas de transporte activo (bomba Na/K)
- Ejercicios de potencial de acción completo

- Casos clínicos de alteraciones electrolíticas
- Problemas de cinética de canales
- Ejercicios de sinapsis y neurotransmisión

Formato JSON sugerido:

```
{
  "id": "osm_003",
  "title": "Título del problema",
  "category": "osmosis",
  "difficulty": 3,
  "statement": "Enunciado completo...",
  "given_data": {
    "variable": {"value": 100, "unit": "mM"}
  },
  "solution": {
    "steps": [
      {"step_number": 1, "description": "...", "formula": "...", "calculation": "..."}
    ],
    "final_answer": {"value": 285, "unit": "mOsm/L"},
    "interpretation": "Explicación del significado..."
  },
  "related_solver": "osmolarity_comparison",
  "tags": ["ósmosis", "tonicidad", "NaCl"]
}
```

Módulos Futuros — Versión 2.0

13. Módulo de Cinética Enzimática

Funcionalidades propuestas:

- Ecuación de Michaelis-Menten
- Gráficos de Lineweaver-Burk y Eadie-Hofstee
- Tipos de inhibición (competitiva, no competitiva, acompetitiva)
- Simulación de reacciones enzimáticas
- Determinación de Km y Vmax

14. Módulo de Termodinámica Biológica

Funcionalidades propuestas:

- Energía libre de Gibbs (ΔG)
- Equilibrio químico y constante de equilibrio
- Acoplamiento energético
- ATP como moneda energética
- Reacciones exergónicas y endergónicas

15. Módulo de Biofísica de Radiaciones

Funcionalidades propuestas:

- Decaimiento radiactivo (vida media)
- Dosis y exposición

- Efectos biológicos de la radiación
- Protección radiológica básica

16. Sistema de Progreso del Estudiante

Funcionalidades propuestas:

- Registro de problemas resueltos
- Estadísticas de uso por módulo
- Sistema de logros y medallas
- Modo de práctica con tiempo
- Historial de cálculos recientes

🛠 Mejoras Técnicas

17. Testing Automatizado

Estado: No implementado

Tareas:

- Tests unitarios para todos los solvers
- Tests de integración para repositorios JSON
- Mocks para testing de GUI
- CI/CD con GitHub Actions
- Cobertura de código objetivo: >80%

Estructura propuesta:

```
tests/
├── unit/
│   ├── test_osmosis_solvers.py
│   ├── test_nernst_solver.py
│   ├── test_iv_curve_solver.py
│   └── test_repositories.py
└── integration/
    └── test_solver_service.py
└── conftest.py
```

18. Documentación del Código

Estado: Docstrings básicos

Tareas:

- Generar documentación con Sphinx o MkDocs
- Añadir ejemplos de uso en docstrings
- Crear guía de contribución (CONTRIBUTING.md)
- Documentar API de solvers

19. Optimización de Rendimiento

Tareas:

Ventajas:

- Lazy loading de vistas (cargar solo al navegar)
- Caché de cálculos frecuentes
- Optimizar renderizado de gráficos grandes
- Reducir tiempo de inicio de la aplicación

20. Empaquetado Mejorado

Estado: PyInstaller básico

Tareas:

- Crear instalador para Windows (Inno Setup)
- Incluir ícono y metadatos de aplicación
- Crear versión portable sin instalación
- Auto-actualización desde repositorio
- Firmar digitalmente el ejecutable

Migración Web — Versión 3.0

21. API REST Backend

Tecnología propuesta: FastAPI + Python

Endpoints:

```
POST /api/v1/osmosis/compare
POST /api/v1/osmosis/volume
POST /api/v1/equilibrium/nernst
POST /api/v1/equilibrium/goldman
POST /api/v1/patch-clamp/iv-curve
POST /api/v1/patch-clamp/single-channel
GET /api/v1/conferences
GET /api/v1/bibliography
GET /api/v1/problems
```

22. Frontend Web

Tecnología propuesta: React + TypeScript + TailwindCSS

Ventajas:

- Acceso desde cualquier dispositivo
- No requiere instalación
- Actualizaciones automáticas
- Posibilidad de modo colaborativo
- Integración con LMS (Moodle, etc.)

Checklist de Release

Antes de distribuir una nueva versión, verificar:

- Todos los solvers retornan resultados correctos
 - La GUI no tiene errores visibles ni warnings
 - Los PDFs de conferencias, bibliografía y seminarios se abren
 - El empaquetado con PyInstaller funciona en Windows 10/11
 - El README.md está actualizado
 - Se han actualizado los números de versión en `config.py`
 - Los datos de ejemplo están incluidos
 - El logo aparece correctamente en sidebar, home y taskbar
-

Convenciones de Desarrollo

Estilo de Código

- **Type hints:** Obligatorios en funciones públicas
- **Docstrings:** Formato Google, en español
- **Nombres de variables:** En inglés (snake_case)
- **Mensajes de usuario:** En español
- **Linter:** Pyright configurado en `pyrightconfig.json`

Estructura de Commits

```
feat: añadir nueva funcionalidad
fix: corregir error
docs: actualizar documentación
style: cambios de formato (sin cambio de lógica)
refactor: reestructurar código
test: añadir o modificar tests
chore: tareas de mantenimiento
```

Flujo de Trabajo Git

1. Rama principal: `main`
 2. Ramas de feature: `feature/nombre-descriptivo`
 3. Ramas de fix: `fix/descripción-del-bug`
 4. Pull Request con revisión antes de merge
-

Contribución

Para contribuir al proyecto:

1. Fork del repositorio
2. Crear rama: `git checkout -b feature/nueva-funcionalidad`
3. Desarrollar con tests
4. Commit: `git commit -m 'feat: descripción concisa'`
5. Push: `git push origin feature/nueva-funcionalidad`
6. Crear Pull Request con descripción detallada



Contacto

Autores:

- Ana Gabriela Zaragoza Palmarola — anagabrielazaragozapalmarola@gmail.com
- Sebastián González — sebagonz106@gmail.com

Repositorio: github.com/sebagonz106/biophysics-virtual-environment

Este documento debe actualizarse conforme se implementen mejoras.